TU WIEN Informatics

Addressing Data Availability and Document-to-Document Retrieval for Domain-specific Neural Rankers

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktorin der Technischen Wissenschaften

by

Sophia Althammer, BSc, Msc
Registration Number 12038077

to the Faculty of Informatics

at the TU Wien

Advisor: Univ. Prof. Dr. Allan Hanbury
Second advisor: Dr. Suzan Verberne

The dissertation has been reviewed by:

Udo Kruschwitz	Gianmaria Silvello

Vienna, 28th November, 2023

Sophia Althammer

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Sophia Althammer, BSc, Msc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 28. November 2023

Sophia Althammer

Kurzfassung

Neuronale Ranking- und Retrieval-Modelle, die auf vortrainierten Sprachmodellen basieren, haben im Vergleich zu statistischen und frühen neuronalen Ranking-Modellen im Bereich der Websuche große Effizienzgewinne gezeigt. Die Übertragung dieser Fortschritte auf domänenspezifische Retrieval-Aufgaben stellt die neuronalen Ranking- und Retrieval-Modelle vor mehrere Herausforderungen: die Fragen und Dokumente können länger sein als bei Websuche, und für das domänenspezifische Retrieval sind im Vergleich zur Websuche weniger hochwertige Evaluierungs- und Trainingsdaten verfügbar. In dieser Arbeit befassen wir uns mit diesen Herausforderungen mit dem Ziel, die Adaption von neuronalen Ranking- und Retrievalmodellen für domänenspezifische Retrievalaufgaben zu fördern und zu verbessern. Dokument-zu-Dokument Retrieval-Aufgaben, bei denen die Fragen und die Dokumente im Korpus lange Dokumente sind, sind wichtige Aufgaben im Rechts- und Patentbereich. Wir reproduzieren und verbessern ein Interaktionsmodell auf Paragraphenebene für die Dokument-zu-Dokument-Suche in der Rechtsdomäne und demonstrieren die Effektivität der Modelle für die Suche nach Prior Art Search im Patentbereich. Um die Verbesserungen der ersten Stufe der Retrieval-Methoden aus der Websuche auf die Aufgabe der Dokument-zu-Dokument-Suche zu übertragen, schlagen wir ein Passagenaggregationsmodell vor. Das Passagenaggregationsmodell befreit neuronale Retrievalmodelle für Retrieval in der ersten Stufe von ihrer begrenzten Eingabelänge und erhöht die Effektivität und Interpretierbarkeit für die Aufgabe des Retrievals von Rechtsfällen. Wir verbessern die Verfügbarkeit von qualitativ hochwertigen Evaluierungsdaten, indem wir eine Annotationskampagne durchführen und Relevanzsignale aus den Klickdaten mit unseren menschlichen Annotationen für die domänenspezifische Suche in der Gesundheitsdomäne vergleichen. Da annotierte Trainingsdaten für domänenspezifische Retrieval-Aufgaben begrenzt und teuer zu erstellen sind, untersuchen wir das Training neuronaler Ranking- und Retrieval-Modelle mit einem begrenzten Annotations- und Trainingsbudget. Wir untersuchen, inwieweit aktive Lernmethoden die Annotationseffizienz für das Training von neuronalen Ranking- und Retrievalmodellen verbessern, wobei wir uns auf eine kostenbasierte Bewertung konzentrieren.

Abstract

Neural ranking and retrieval models based on pretrained language models have demonstrated great effectiveness gains for Information Retrieval (IR) in the web domain compared to statistical and early neural ranking models. Bringing these advancements to domain-specific retrieval tasks poses multiple challenges for neural ranking and retrieval models: the queries and documents can be much longer than in web search and there is less high-quality evaluation and training data available for domain-specific retrieval compared to web search.

In this thesis we address these challenges with the goal of promoting and improving the adoption of neural ranking and retrieval models for domain-specific retrieval tasks. Document-to-Document retrieval tasks, where the query and the documents in the corpus are long documents, are important tasks in the legal and patent domain. We reproduce and improve a paragraph-level interaction re-ranking model for the document-to-document retrieval task of legal case retrieval and we demonstrate the re-ranking models' effectiveness for prior art search in the patent domain. In order to bring improvements of first stage retrieval methods from web search to the task of document-to-document retrieval, we propose a paragraph aggregation retrieval model. The paragraph aggregation retrieval model liberates neural first stage retrieval models from their limited input length and increases effectiveness and interpretability in the first stage retrieval for the task of legal case retrieval.

We increase the availability of high-quality evaluation data by conducting an annotation campaign and comparing relevance signals from the click data to our human-label annotations for domain-specific retrieval in the health domain. Since annotated training data is limited and expensive to produce for domain-specific retrieval tasks, we study training neural ranking and retrieval models under a limited annotation and training budget. We investigate active learning methods for improving the annotation efficiency for training neural ranking and retrieval models focusing on a cost-based evaluation.

Contents

CHAPTER 1

Introduction

Search engines are omnipresent, processing over 3 billion queries and serving millions of users each day [Goo]. There are many different ways in which search engines assist us: for planning a trip, for entertainment, or for executing work tasks. Especially for executing work tasks, search engines are an essential tool for professionals in various domains, particularly for those whose duties involve retrieval tasks as a core component of their work [RRCA18].

As **Search Example ❶** let us consider an attorney in the United States, whose task at work is to defend a client in a trial. Since in the United States the legal system relies on precedent legal cases, the attorney needs to find already decided precedent cases, which are similar to the case of his client, in order to develop a defense strategy for this case. Here it is crucial, that the attorney finds all precedent cases, that are somehow related to the clients case, in order to develop the best possible defense strategy and to be prepared for possible arguments of the opposite lawyer. Here the lawyer can use search engines to collect necessary evidence, provide guidance, and develop a defense strategy. Let us assume that the attorney queries the search engine with the description of the current client case, in order to find other similar legal cases [RKG+20]. This example is an example for prior case retrieval in the legal domain.

As **Search Example ❷** let us consider a patent attorney, whose task is to write a patent application for a novel invention. This involves checking the current state-of-the-art of granted patents, that are related to the patent application about the novel invention. Thus the patent attorney needs to find related patents, that have already been granted and check if the novel patent application is significantly different and novel compared to the existing patents and if needed re-write the patent application, so that the patent application does not infringe on existing, granted patents. Here it is important, that the patent attorney finds all related, already granted patents, so that he can differentiate the novel patent application from the state-of-the-art and if needed cite related, already granted patents in the novel patent application. Here the patent attorney can use search engines to collect existing patents and compare them to the novel patent application. Let us assume that the patent examiner queries the search engine with the novel patent application, in

order to find other related patents [PLHZ11]. This example is an example for prior art search in the patent domain.

As **Search Example ❸** let us consider a medical doctor in the United Kingdom, who is working in an emergency department of a hospital and needs to find a medical treatment for a patient. His task is to find the best possible treatment for his patient considering the patients' health condition and patients' characteristics like age, gender or medical history as well as the state-of-the-art of clinical research results. Furthermore his decisions are time-critical, thus he needs to find the best treatment strategy quickly. Here the medical doctor can use a search engine to collect evidence, develop a treatment strategy and make critical decisions. Let us assume that the medical doctor queries the search engine with short, medical terms describing the patients health condition, in order to find medical treatments or clinical trials [RLS+21]. These are two examples of domain-specific search, that will guide us through this thesis. This example is an example for ad-hoc retrieval in the web domain.

We want to define the scope of domain-specific search in this thesis. We follow the definition of Lupu et al. [LSH14] for domain-specific search, who define a domain-specific search [engine or process] as a search [engine or process] that specifies one or more of the following five dimensions:

1. subject areas/domains e.g. legal, patent, medical
2. modality e.g. text, images, videos, sounds
3. users e.g. a paralegal, a patent examiner, a doctor
4. tasks e.g. prior legal case retrieval, prior art patent search, health ad-hoc retrieval
5. tools, techniques and algorithms required to complete the tasks, e.g. query completion limited to specific vocabularies, cross-lingual search, possibility to store search results

This definition shows that domain-specific search can be characterized by different aspects: characteristics of the information sources (the subject area or modality), characteristics of the users (the users or tasks) or technical aspects (tools, techniques and algorithms) being domain-specific. We do not limit this definition of domain-specific search to search with professional users e.g. users who conduct search in a work context [VHW+19]. When users conduct search in a work context and the users are paid professionals, this is defined as professional search [Tai14, VHW+19, RRCA18, KH17]. We consider professional search as domain-specific search if the search fulfills one of the above mentioned properties, however in domain-specific search the users can be professionals and also layperson.

Search Example ❶ fulfills multiple of the above mentioned characteristics: the search process is in the legal domain, the user is a legal professional and the task is a specific one e.g. prior case retrieval in the legal domain. In **Search Example ❷** the search process also fulfills multiple characteristics from the above definition. The search process is in the patent domain, the user is a professional patent attorney and the task is prior art search. In **Search Example ❸** the search process is subject to the medical domain, the user is a medical professional and the task is health ad-hoc retrieval.

Due to the many stakeholders in industry, government and research, who face domain-specific retrieval tasks on a daily basis [LSH14], studying domain-specific retrieval tasks is at the core of Information Retrieval (IR) research and has a long history within the community [HTBO09a, GCR16, LSH14]. Improving retrieval systems for domain-specific retrieval tasks benefits a variety of stakeholders and can leverage more effective and efficient execution of work tasks [RRCA18]. In a world with an exponentially growing amount of information [Num] and with an increasing amount of information tasks in the workplace, research on domain-specific retrieval tasks becomes even more important and is crucial for the future development of IR research.

With the advent of large pre-trained language models [DCLT19] based on Transformers [VSP+17] in the Natural Language Processing community and their powerful capabilities of representing and contextualizing text, their application in the context of ranking and retrieval systems is natural. Ranking and retrieval models based on large pre-trained language models, which we also refer to as neural ranking and retrieval models or neural rankers, have shown massive effectiveness gains for retrieval tasks in the web domain [NC19, CMYC19]. The neural models can be categorized into ranking (also re-ranking) and retrieval models by their computational complexity [KOM+20, KZ20, NC19]. Theoretically neural ranking models could also be used to score the whole collection and retrieve relevant documents, but due to the models' high computational complexity and thus the models' long inference time at query time, the neural re-ranking models are used to re-rank a list of top N documents, which are retrieved in a first stage by a much more efficient retrieval model [NC19].

The great potential of large pre-trained language models and the continuous research on their capabilities [CRM+22, Ope23] and on how to employ them most effectively for ranking and retrieval [HLY+21, DZM+23], will make neural ranking and retrieval models even more effective in the future.

Neural ranking and retrieval models hold the promise to bring the demonstrated advancements in the web domain also to domain-specific retrieval tasks. The powerful contextualization mechanisms [VSP+17] and the immense pre-training of the neural models [DCLT19] address shortcomings of currently existing lexical or shallow neural retrieval models [RZ09]. Given the large improvements of neural ranking and retrieval models in the web domain [CMYC19, CMYC20, NC19], we hypothesize that the adaptation of neural ranking and retrieval models for domain-specific retrieval has a great potential for boosting retrieval effectiveness. Bringing the large effectiveness gains of neural ranking and retrieval models from the web domain to domain-specific retrieval tasks, is also crucial for the democratization of neural ranking and retrieval models so that not only large corporations in web search, but also stakeholders of domain-specific retrieval tasks benefit from the research advancements.

Domain-specific retrieval tasks are challenging and difficult retrieval problems, due to the domain-specific information workflows during the task [Kuh91, KT01], the domain-specific information needs of the users [LSH14, FEL19], and the domain-specific notion of relevance [vOS17]. Domain-specific retrieval tasks may have very different characteristics than tasks in the web domain: domain-specific retrieval tasks may have a different notion of relevance than relevance in web search [vOS17]; the tasks may inherit different information needs than web search for example high-recall search [LSH14, FEL19, RGK21] like our **Search Example ❶** or **Search**

Example ❷; tasks may have different characteristics like long queries and long documents in the collection [RKG+20, PH19] like our **Search Example ❶** or **Search Example ❷**; and the user queries and documents may be written in domain-specific language containing domain-specific terms and classification schemes [RLS+21, RKG+20, PH19] like our **Search Example ❸**. Retrieval systems need to be designed and evaluated along the domain-specific tasks' workflows and characteristics [SW21], in order to fulfill the information needs of the users. The different characteristics of domain-specific retrieval tasks lead to challenges, when employing neural ranking and retrieval models for those tasks: to learn and evaluate the different notion of relevance, large scale training and reliable evaluation sets are necessary; different metrics than in web search need to be employed to evaluate if the retrieval systems fulfill the information need; different model architectures for long queries and documents are necessary; and different pre-trained language models are necessary to contextualize the domain-specific language effectively. To attain highly effective domain-specific retrieval models, it is necessary to investigate, how neural ranking and retrieval models can be reliably evaluated, effectively and efficiently trained and adapted for domain-specific retrieval tasks.

1.1 Open Challenges

In this thesis we address two main challenges of neural ranking and retrieval models for domain-specific tasks: handling long queries and long documents, which we refer to as document-to-document retrieval tasks, and availability of evaluation and training data.

Document-to-document retrieval tasks, also referred to as query-by-example tasks [AVA22] or extremely long queries and documents [AVA+23], are retrieval tasks where the query and the items in the collection to be retrieved are long documents. With long documents we refer to documents that greatly exceed the average length of queries or web pages as in the web domain [HVCB99]. There are various domain-specific document-to-document retrieval tasks for example in the legal [RKG+20], the patent [NFIH10], and the scientific domain [CFB+20]. The active research on those tasks with evaluation campaigns [PLHZ11] and competitions [RKG+20] demonstrates the importance of the tasks within the IR community and for the stakeholders in the industry.

Our **Search Example ❶** is an example for a document-to-document retrieval task in the legal domain. Here the query is the current legal client case, which is a long document, and the collection consists of precedent legal cases, which are also long documents. Furthermore **Search Example ❷** is also an example for a document-to-document retrieval task, but in the patent domain. The query is the novel patent application, thus a long document, and the documents in the collection are granted patents, that are also long documents.

Incorporating the whole content of the query and the documents for neural re-ranking shows large effectiveness gains for prior case retrieval in the legal domain [SML+20], thus it is a promising and important direction to investigate these findings for their generalizability for document-to-document retrieval tasks in other domains. Furthermore it is an open question, how we can adapt neural first stage retrieval models for document-to-document retrieval tasks, so that they effectively take the whole content of the query and document into account.

The availability of data for both training and evaluation is a primary obstacle for researching domain-specific, neural ranking and retrieval models. Data availability is crucial for evaluating and comparing any model to another as well as for training a model. Even the retrieval effectiveness of lexical retrieval models, which do not need to be trained, benefit from training data to be used for fine-tuning its input parameters [RZ09]. Thus the lack of data is a bottleneck to expand research on domain-specific neural ranking and retrieval models in the Information Retrieval community. Especially for neural ranking and retrieval models, their high effectiveness relies on large-scale, human-labelled training data [CMYC19, CMYC20]. Without additional investigation it is not clear, how well old, domain-specific test collections are suited to evaluate novel neural ranking and retrieval models [VSL22], when neural ranking and retrieval models did not participate in the pooling process. Thus for evaluating neural ranking and retrieval models in the context of domain-specific search, it is crucial to have reliable and reusable evaluation data at hand, where neural ranking and retrieval models contributed to the pool to be judged. For domain-specific retrieval tasks, we lack large-scale, human-labelled training and reliable evaluation sets for the various tasks within those domains. Furthermore it is highly expensive create such training or evaluation sets, since the annotation of the samples needs to be done by domain experts with high hourly rates and the relevance assessments usually take more time than in web search [AHVH22].

Since reliable evaluation data is rare and very expensive to produce for domain-specific retrieval tasks, some common, domain-specific test collections rely on relevance signals from user behaviour [RLS$^+$21, TRR$^+$21]. It is an open research gap, how reliable domain-specific test collections, which do not rely on relevance judgements from pooled system rankings, are for evaluating retrieval systems that did not take part in the pooling for relevance.

Our **Search Example ❸** is an example for a domain-specific retrieval task in the medical domain with little, human-annotated evaluation data [RDVH16, RLS$^+$21]. Whereas a variety of evaluation campaigns exist in the medical domain [CMS21, RDV$^+$22, RDV$^+$19, RDVH16], these evaluation campaigns often focus on specific aspects of ad-hoc retrieval in the medical domain e.g. health misinformation [CMS21] or cancer patients [RDV$^+$19] or are other retrieval tasks e.g. systematic reviews [**?**].

Large-scale, high-quality training data is necessary for trained neural ranking and retrieval models to perform well [CMYC19, CMYC20]. Since it is costly to annotate training samples for domain-specific retrieval tasks, it is an open research gap how we can effectively train neural ranking and retrieval models under a limited cost budget. It is an open, but important question, how we can minimize the annotation cost while optimizing the effectiveness of the trained neural ranking or retrieval model using active learning methods. Active learning methods hold the promise to minimize the number of annotations of the training samples while maximizing the effectiveness of the trained model by selecting iteratively which training samples to annotate, following a pre-defined selection strategy [CGJ96].

1.2 Research Questions

We bridge the above research gaps by addressing the following research questions:

RQ1 How can neural ranking and retrieval models be adapted for document-to-document retrieval tasks?

We divide this research questions into neural ranking approaches and neural retrieval models and investigate:

RQ1.1 How can neural ranking models be adapted for document-to-document retrieval tasks?

We study how neural ranking models proposed for ad-hoc retrieval in the web domain [NC19] can be adapted for document-to-document retrieval tasks. We specifically study the document-to-document retrieval tasks of prior case retrieval in the legal domain and prior art search in the patent domain. Here we reproduce the novel neural re-ranking architecture BERT-PLI, which was proposed for prior case retrieval [SML+20], and investigate the transferability of BERT-PLI for the task of prior art search in the patent domain [PLHZ11]. Thus we investigate the following sub research questions:

RQ1.1.1 Does fine-tuning BERT on domain specific paragraphs improve the retrieval performance for document retrieval?

Since we reproduce Shao et al.'s [SML+20] work, we re-investigate their findings and study if fine-tuning the BERT-PLI model for modelling the paragraph interactions on domain specific paragraphs improves the overall retrieval performance of the BERT-PLI model.

RQ1.1.2 To what extent is a BERT-PLI model, which is trained on patent retrieval, beneficial for document retrieval in the patent domain?

In addition to the reproduction, we explore how effective the BERT-PLI model is for the task of prior art search in the patent domain, which is also a document-to-document retrieval task.

RQ1.1.3 To what extent is cross-domain transfer on paragraph- and document-level of the domain specific BERT-PLI model between legal and patent domain possible?

Furthermore we investigate the effectiveness of cross-domain transfer of the domain-specific BERT-PLI models. Here we evaluate the BERT-PLI model, trained on the prior legal case retrieval task, for prior art search in the patent domain and vice versa. With this we study, to what extend we can transfer the neural ranking model across the legal and patent domain.

RQ1.2 How can neural retrieval models be adapted for document-to-document retrieval tasks?

We investigate how we can adapt neural retrieval models, proposed for ad-hoc retrieval in the web domain [KOM+20], for document-to-document retrieval tasks, where we also investigate prior case retrieval in the legal domain and prior art search in the patent domain. For this we

propose a novel paragraph-aggregation retrieval model (PARM), which liberates neural retrieval models from their limited input length. PARM retrieves documents on the paragraph-level and then aggregates the relevant results per query paragraph into one ranked list for the whole query document. For the aggregation we propose vector-based aggregation with reciprocal rank fusion (VRRF) weighting, which is an aggregation approach that combines rank-based aggregation of results [CCB09] and topical aggregation based on the neural embedding. Thus we investigate:

RQ1.2.1 How does VRRF compare to other aggregation strategies within PARM?

We compare our novel aggregation strategy VRRF to other aggregation strategies [CCB09, Lee97, SF94] for the paragraph-aggregation retrieval model (PARM) and study which aggregation strategy leads to the highest effectiveness for the task of legal case retrieval. Furthermore we study:

RQ1.2.2 How effective is PARM with VRRF for document-to-document retrieval?

We train and evaluate the paragraph-aggregation retrieval model (PARM) for the task of legal case retrieval and compare its effectiveness to other neural and non-neural first stage retrieval models. Here we investigate the effectiveness of taking into account the whole query document and the whole document in the corpus compared to only using a smaller text representation of the query or the document.

RQ1.2.3 How can we train neural retrieval models for PARM for document-to-document retrieval most effectively?

Since freely, available training data for the domain-specific retrieval task of legal case retrieval is limited [RKG+20], we investigate how we can train neural retrieval models for PARM most effectively. Here we compare the effect of training on paragraph-level training data only or additionally training on document-level training data on the effectiveness of PARM with the neural retrieval model.

RQ2 How can the problem of limited available annotated evaluation and training data be addressed in domain-specific retrieval?

In order to address the problem of limited available annotated evaluation and training data, we investigate two possible directions. First, we conduct an annotation campaign, in which human annotators annotate an evaluation set, that previously only contained relevance labels based on click-signals. With this annotation campaign, we create a human-annotated evaluation set for the task of medical ad-hoc retrieval. We investigate:

RQ2.1 How do human-label annotations compare to click signals for medical ad-hoc retrieval?

We compare our human-annotated evaluation set for medical ad-hoc retrieval with the click-based labels from the original evaluation set quantitatively and qualitatively, in order to investigate the

difference of the annotations to the click-labels. We also investigate the effect of the different labels on the evaluation of different retrieval and ranking systems.

In order to address the problem of limited available training data for training domain-specific neural ranking and retrieval models, we investigate:

RQ2.2 To what extent does active learning improve annotation efficiency for training neural ranking and retrieval models?

Active learning or active selection strategies are strategies to select and annotate training samples, that aim to minimize the number of annotations needed and maximize the effectiveness of the model trained on that annotated training set [CGJ96]. We study how active learning methods influence the annotation efficiency for annotating the training set, when training neural ranking and retrieval models. Since we want to disentangle the effects of active learning for efficient training data annotation from possible effects of training domain-specific neural ranking and retrieval architectures like BERT-PLI or PARM, we investigate these active learning strategies for "common" neural ranking and retrieval model architectures [NC19, KOM+20, KZ20], that do not consider long documents. For investigating training neural ranking and retrieval models, we investigate two scenarios. First, training the neural ranking or retrieval model from "scratch", where the training starts with a pre-trained large language model that is not trained on a ranking or retrieval task yet. Second, adapting a neural ranking or retrieval model, that is already trained on an ad-hoc retrieval task in the web domain, to ad-hoc retrieval in the health domain. In order to investigate the effects of adapting the language domain, we choose to investigate a similar task, that exists in the web as well as in the health domain (ad-hoc retrieval). Another reason for choosing the tasks of ad-hoc retrieval in the web domain and the medical domain when investigating active learning methods for data efficient training of neural rankers, is the availability of large-scale, annotated training data for these two tasks [NRS+16, RLS+21], which we need, in order to simulate the selection and annotation process in the training. We divide the above research question in multiple sub research questions and investigate:

RQ2.2.1 What is the effect of the size of the labelled training data on the effectiveness of neural rankers?

Since investigate training neural ranking and retrieval models under a limited annotation and training budget, we study the effect of the size of the labelled training data on the effectiveness of the neural ranking and retrieval model, that is trained on different sizes of training data. Here we study training neural ranking and retrieval models from scratch for ad-hoc retrieval in the web domain or adapting neural ranking and retrieval models, already trained for web ad-hoc retrieval, to the task of ad-hoc retrieval in the medical domain. Then we investigate:

RQ2.2.2 How do different active selection strategies influence the effectiveness of neural rankers?

We adapt active selection strategies, that were proposed for classification [XAZ07, LG94] or non-neural ranking models [FSST97], for neural ranking and retrieval models and study their influence on training neural rankers. We evaluate their effectiveness for different training datasizes

for ad-hoc retrieval in the web and the health domain. In order to conduct a cost-aware evaluation, we investigate:

RQ2.2.3 What is the effect of using an an active selection strategy to fine-tune a neural ranker under a constrained budget?

We propose a cost-aware evaluation, that takes into account the annotation cost as well as the training cost of training the neural rankers. Furthermore we measure the annotation and training cost for training neural rankers with different active selection strategies and compare the annotation and training cost to the effectiveness of the trained neural rankers. With answering these research questions, we overall investigate how to train neural rankers under a limited annotation and training budget and address the problem of limited training data for training neural ranking and retrieval models.

1.3 Thesis Contributions

Along our main research questions, our contributions are the following:

RQ1 How can neural ranking and retrieval models be adapted for document-to-document retrieval tasks?

First we investigate for neural ranking models:

RQ1.1 How can neural ranking models be adapted for document-to-document retrieval tasks?

In order to investigate how neural re-ranking models can be adapted for document-to-document retrieval tasks, we reproduce the experiments by Shao et al. [SML+20], who propose a paragraph-level interaction re-ranking model for legal prior case retrieval. We make our reproduction code publicly available at the following URL [1] as well as the models under the following repository [Alt20]. Contrary to the original paper we find that domain-specific paragraph-level modelling does not benefit the effectiveness of the neural re-ranking model for the task of prior case retrieval in the legal domain. We extend the work by Shao et al. [SML+20] by training the proposed model architecture for the document-to-document retrieval task of prior art search in the patent domain and investigate the generalizability of the findings to the task of prior art search. Furthermore we evaluate cross-domain transfer of the re-ranking models and find first promising results.

RQ1.2 How can neural retrieval models be adapted for document-to-document retrieval tasks?

For a high-recall, first-stage retrieval of document-to-document retrieval tasks we propose a paragraph aggregation retrieval model (PARM) for neural document-to-document retrieval and make the code and trained models publicly available under the following URL [2]. PARM liberates neural retrieval models from their limited input length and thus makes it possible to apply neural

[1] https://www.github.com/sophiaalthammer/bert-pli
[2] https://www.github.com/sophiaalthammer/parm

retrieval models for document-to-document retrieval including the whole content of the query and the document. In order to aggregate the retrieved paragraphs, we propose vector-based aggregation with reciprocal rank fusion weighting (VRRF) for neural retrieval with PARM and find that VRRF leads to the highest recall for PARM compared to other common aggregation strategies [SF94, CCB09]. In our experiments we demonstrate higher retrieval effectiveness for neural retrieval with PARM compared to retrieval without PARM and to lexical retrieval with PARM for the task of prior case retrieval in the legal domain. Furthermore we investigate PARM for the task of prior art search in the patent domain and find that neural retrieval models are not yet beneficial for this task, both with the PARM architecture and with off-the-shelf neural retrieval with limited input length.

RQ2 How can the problem of limited available annotated evaluation and training data be addressed in domain-specific retrieval?

When creating an evaluation set through an annotation we campaign, we investigate:

RQ2.1 How do human-label annotations compare to click signals for medical ad-hoc retrieval?

To increase the availability of evaluation data in the medical domain, we create the relevance judgement-based test collection TripJudge for TripClick health retrieval and make it publicly available under the following URL [3]. We ensure the quality and re-usability of TripJudge by a variety of statistical and neural ranking and retrieval systems for pool creation, by multiple judgements per query-document pair, and by an at least moderate inter-annotator agreement in our large-scale annotation campaign. We compare evaluation with click-based TripClick and our judgment-based TripJudge and find that click and judgment-based evaluation lead to substantially different system rankings.

RQ2.2 To what extent does active learning improve annotation efficiency for training neural ranking and retrieval models?

We investigate to what extent active learning methods improve the annotation efficiency for training neural ranking and retrieval models. In order to distinguish the impact of active learning on the annotation efficiency of training data from any potential influences related to domain-specific neural ranking and retrieval architectures such as BERT-PLI or PARM, we are examining these active learning methods within the context of "common" neural ranking and retrieval model architectures [NC19, KOM+20, KZ20], which are not designed to handle lengthy documents. To explore the training of neural ranking and retrieval models, we are investigating two distinct scenarios. Firstly, there is the approach of training the neural ranking or retrieval model "from scratch," which entails initiating training with a pre-trained large language model that has not previously been fine-tuned for ranking or retrieval tasks. Here we study training a neural ranker under a limited annotation and training budget for the task of ad-hoc retrieval in the web domain. Secondly, we are examining the process of adapting a neural ranking or retrieval model, that has

[3] https://www.github.com/sophiaalthammer/tripjudge

already been trained for ad-hoc retrieval in the web domain, to the specific context of ad-hoc retrieval in the medical domain. Our goal in choosing this adaptation scenario is to assess the impact of transferring the model across different language domains, as we are working with a similar task, that is applicable in both web and health domains, which is ad-hoc retrieval. One additional factor in the selection of ad-hoc retrieval tasks within the web and medical domains for our exploration of active learning methods for training neural rankers is the availability of large-scale, annotated training data for these specific tasks [NRS+16, RLS+21]. This annotated training data is needed for simulating the process of selection and annotation during training, since we do not have the resources, to do an interactive annotation campaign during the selection and training process. We adapt active learning strategies [LG94, XAZ07, CGZW11] for training neural ranking or retrieval models and propose a budget-aware evaluation schema including aspects of annotation and computational cost. We conduct an extensive analysis of active learning strategies for training neural ranking and retrieval models investigating the trade-offs between effectiveness, annotation budget and computational budget. Interestingly, we find that no active learning method consistently and significantly outperforms random selection of training data for annotation and training. However, we find that some subsets of the training data result in considerably higher effectiveness than others. Our budget-aware evaluation shows that the investigated active learning strategies do not deliver consistent budget savings.

1.4 Synopsis

We give an overview of the outline of the thesis by describing each chapter and then list the publications that some of the chapters are based on. In these chapters, the results are presented as they were at the time of publication, there are no new baselines added retrospectively.

- In the first chapter of the thesis we give an introduction we motivate our research questions, outline open challenges in section 1.1 and introduce the research questions in section 1.2 that are addressed in this thesis. Furthermore we state the contributions of this thesis on top of the current research landscape in section 1.3.

- In the second chapter we give the background about the domains in section 2.3 and their respective retrieval and ranking tasks in section 2.4, that we study in this thesis. Furthermore we introduce the datasets in section 2.5 and model architectures that will be used.

- In the third chapter of the thesis, we lay out the related work for document-to-document retrieval in section 3.1 as well as the challenge of lack of data for domain-specific retrieval tasks in section 3.2.

- In the fourth main part of this thesis, we describe our approach to investigate, how to adapt neural ranking and retrieval models for document-to-document retrieval tasks. In particular we study the task of legal case retrieval in the legal domain and the task of prior art search in the patent domain and focus on a high-recall evaluation for these tasks. In section 4.1 we investigate how neural re-ranking models can be adapted for document-to-document retrieval tasks by reproducing the experiments by Shao et al. [SML+20] and extending the

experiments for the task of prior art search. In section 4.2 we investigate the adaptation of neural first stage retrieval models for document-to-document retrieval tasks. We propose a paragraph aggregation retrieval model (PARM) for neural document-to-document retrieval and study its effectiveness for the tasks of legal case retrieval and prior art search.

- In the fifth main part of this thesis, we address the problem of limited available annotated evaluation and training data for domain-specific retrieval tasks specifically in the health and web domain. In section 5.1 we address the data availability of reliable evaluation data in the health domain by creating the relevance judgement-based test collection TripJudge for the TripClick health retrieval dataset. Additionally to creating new data resources for domain-specific retrieval tasks, we investigate in section 5.2 how we can train neural ranking and retrieval models under a limited data annotation and training budget. Here we investigate to what extent active learning methods improve the annotation efficiency for training neural ranking and retrieval models for the TripClick health retrieval task as well as for web search. In our investigation, we aim to disentangle the influence of active learning on the efficient annotation of training data from any potential effects introduced by specialized neural ranking and retrieval architectures like BERT-PLI or PARM. To accomplish this, we examine active learning strategies in the context of "common" neural ranking and retrieval model architectures, as identified in the literature [NC19, KOM+20, KZ20]. These architectures are primarily tailored for handling shorter documents and are not explicitly designed for accommodating longer ones. As these neural models are originally proposed for the task of ad-hoc retrieval in web domain and are optimized to handle brief queries and documents that fit within the input length of the BERT encoder [DCLT19], we narrow our focus to web and medical ad-hoc retrieval tasks, when investigating the impact of active learning methods on annotation efficiency. As ad-hoc retrieval tasks are more precision-oriented than recall-oriented, we focus in our evaluation on precision-oriented metrics.

- In the last main chapter 6 we conclude our research findings, describe the contributions to the field as well as we discuss limitations and possible future directions for developing domain-specific neural ranking and retrieval models.

The chapter 4 and 5 are based on four publications that were published in peer-reviewed conferences. We visualize the main focus area of each of the publications (and the respective chapter in the thesis) and the tasks and domains tackled in the publication in Figure 1.1.

The publications that the chapters 4.1, 4.2, 5.1, 5.2 are based on, are the following:

- **Section 4.1** is based on following publication:

2021	**ECIR**	Cross-domain Retrieval in the Legal and Patent Domains: a Reproducibility Study *S. Althammer, S. Hofstätter, A. Hanbury*	[AHH21]

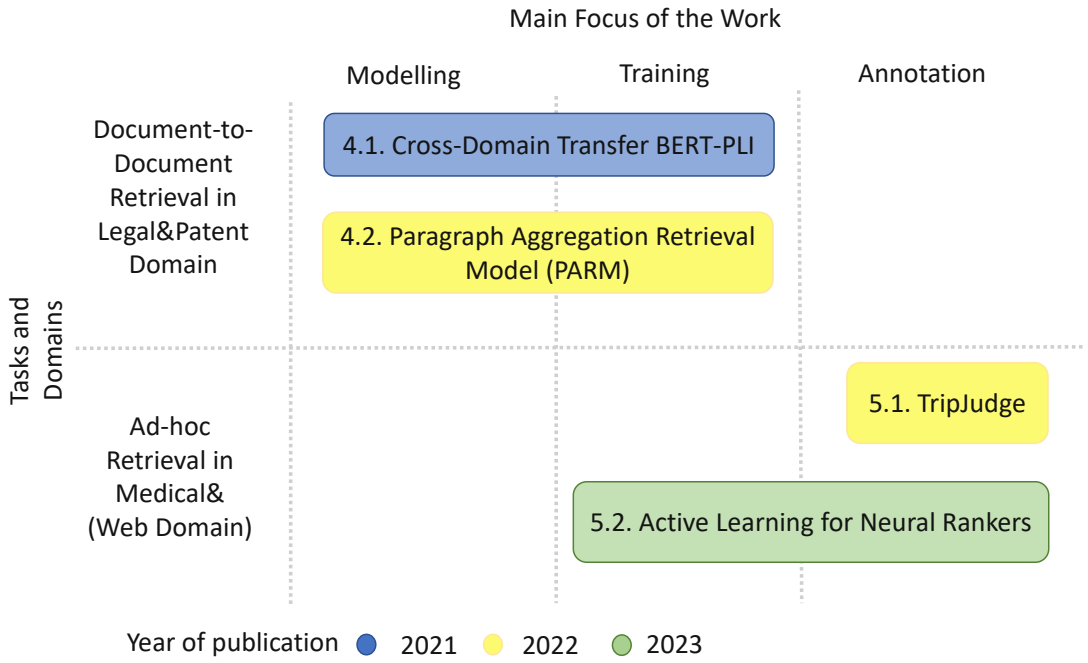


Figure 1.1: Overview of different aspects of our thesis, categorized by tasks and domains, main focus of the work, and year of publication

In this publication we investigate how neural re-ranking models can be adapted for document-to-document retrieval tasks by reproducing the experiments by Shao et al. [SML+20]. Shao et al. propose a paragraph-level interaction re-ranking model for legal prior case retrieval and we investigate shortcomings in the data pre-processing of the original paper and add missing code for reproduction. We extend the work by Shao et al. [SML+20] by training the proposed model architecture for the document-to-document retrieval task of prior art search in the patent domain and investigate the generalizability of the findings to the task of prior art search. Furthermore we propose a cross-domain evaluation approach, in order to evaluate the zero-shot effectiveness of the neural re-ranking model, trained on the prior case retrieval, for prior art search and vice versa. For the effectiveness of the cross-domain transfer of the re-ranking models, we find first promising results.

- Section 4.2 is based on the publication:

2022	**ECIR**	PARM: A Paragraph Aggregation Retrieval Model for Dense Document-to-Document Retrieval *S. Althammer, S. Hofstätter, M. Sertkan, S. Verberne, A. Hanbury*	[AHS+22]

In this publication we investigate the adaptation of neural first stage retrieval models for document-to-document retrieval tasks. We propose a paragraph aggregation retrieval model (PARM) for neural document-to-document retrieval. PARM liberates neural retrieval

models from their limited input length and thus make it possible to apply neural retrieval models for document-to-document retrieval including the whole content of the query and the document. We design and evaluate the PARM architecture along the requirements of the workflow of the retrieval task [SW21], which is a high recall for prior case retrieval in the legal domain and prior art search in the patent domain [RRCA18]. In order to aggregate the retrieved paragraphs, we propose vector-based aggregation with reciprocal rank fusion weighting (VRRF) for neural retrieval with PARM and find that VRRF leads to the highest recall for PARM compared to other common aggregation strategies [SF94, CCB09]. We demonstrate higher retrieval effectiveness for neural retrieval with PARM compared to retrieval without PARM and to lexical retrieval with PARM for the task of prior case retrieval in the legal domain. Furthermore we investigate PARM for the task of prior art search in the patent domain and find that neural retrieval models are not beneficial yet for this task, both with the PARM architecture and with off-the-shelf neural retrieval with limited input length. Overall we propose with PARM a neural first stage retrieval model for document-to-document retrieval, which is more efficient than previous works [SML+20].

- Section 5.1 is based on the publication:

2022	**CIKM**	TripJudge: A Relevance Judgement Test Collection for TripClick Health Retrieval *S. Althammer, S. Hofstätter, S. Verberne, A. Hanbury*	[AHVH22]

In this publication we address the data availability of reliable evaluation data in the health domain by creating the relevance judgement-based test collection TripJudge for TripClick health retrieval. As previous research in the web domain suggests [KKT09], relevance labels from click signals are highly noisy and differ greatly from human-labelled annotations. We re-evaluate this hypothesis in the context of retrieval in the health domain for the TripClick collection. We compare evaluation with click-based TripClick and our judgment-based TripJudge and find that click and judgment-based evaluation can lead to different system rankings.

- Section 5.2 is based on the publication:

2023	**SIGIR-AP**	Annotating Data for Fine-Tuning a Neural Ranker? Current Active Learning Strategies are not Better than Random Selection *S. Althammer, G. Zuccon, S. Hofstätter, S. Verberne, A. Hanbury*	[AZH+23]

In this publication we investigate the question to what extent active learning methods improve the annotation efficiency for training neural ranking and retrieval models. We first investigate how the amount of labelled data used for training the neural ranking or retrieval model impacts its effectiveness and find a great variability in effectiveness when training a neural ranking or retrieval model on different subsets of the same size. We adapt active learning strategies [LG94, XAZ07, CGZW11] to the task of training neural ranking

or retrieval models and propose a budget-aware evaluation schema including aspects of annotation and computational cost. We conduct an extensive analysis of active learning strategies for training neural ranking and retrieval models investigating the trade-offs between effectiveness, annotation budget and computational budget. Interestingly, we find that no active learning method consistently and significantly outperforms random selection of training data for annotation and training. However, we find that some subsets of the training data result in considerably higher effectiveness than others. Our budget-aware evaluation shows that the investigated active learning strategies do not deliver consistent budget savings.

In the following we will use the rhetorical "we" in research when referring to work, which was lead by myself and jointly authored with my co-authors.

Furthermore we will use the terms test collection interchangeably with evaluation set/data/dataset or testing set/data/dataset or test set/data/dataset. Similarly we use the terms training set/data/-dataset. Furthermore we use the term domain-specific search and domain-specific retrieval interchangeably.

CHAPTER 2

Background

In this section we introduce the background in Information Retrieval, evaluation metrics, domains, and their retrieval tasks, which we investigate in this thesis in the context of neural ranking and retrieval models. Furthermore we introduce the datasets, which model the domain-specific retrieval tasks. We give a brief overview of the neural ranking and retrieval models and active learning models. This section is a background needed to understand the related work section as well as the upcoming chapters.

2.1 Information Retrieval

> "Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)." [MRS08]

Manning et al. [MRS08] define information retrieval as the task of finding material from an unstructured source that fulfills the information need of the user. While the users used to be mainly librarians, paralegals, and professional searcher [MRS08] with the emergence of the world wide web now there are billions of users every day search the web.

Information retrieval is a dynamic and evolving research field at the intersection of computer science, information science, and data management [MRS08]. At its core, this discipline is concerned with the efficient and effective retrieval of information from vast and often unstructured data sources, such as text documents, multimedia content, or databases [MRS08]. The overarching goal of information retrieval is to provide users with access to relevant information in response to their queries, facilitating the extraction of knowledge and insights from an ever-expanding digital universe.

In order to explore each aspect and dimension of this Information Retrieval problem, the research in Information Retrieval spans a wide range of techniques and methodologies, including natural

language processing, machine learning, user studies and information retrieval models. These methods aim to enhance the information seeking process, by enabling systems to not only retrieve relevant documents, but also rank them based on their relevance to the user's query or by designing interfaces that help the user navigate in their information seeking process. As our reliance on digital information continues to grow, the field of information retrieval plays a crucial role in shaping, how we access and utilize the wealth of information available to us in the modern era.

2.2 Evaluation Metrics

Evaluation metrics [MRS08] serve as the backbone for assessing the performance and effectiveness of machine learning and information retrieval systems. They play a pivotal role in quantifying, how well these systems accomplish their intended tasks and provide valuable insights for researchers, developers, and end-users. In the realms of machine learning and information retrieval, evaluation metrics are indispensable for comparing and fine-tuning different algorithms, models, and techniques.

These essential metrics help answer critical questions:

- **Accuracy**: How well does the model or system predict or retrieve relevant information, and how often does it do so correctly?
- **Precision and Recall**: In the context of information retrieval, how many of the retrieved documents are relevant (precision), and how many relevant documents were successfully retrieved (recall)?
- **F1 Score**: This metric strikes a balance between precision and recall, particularly useful when precision and recall are in conflict.

The choice of evaluation metrics depends on the nature of the problem. For instance, classification tasks may prioritize metrics like accuracy and F1 score. In information retrieval, the metrics are tailored to the task of retrieving and ranking documents based on relevance. While the above metrics are common metrics also used in Natural Language Processing and general Machine Learning [RKG+20, PLHZ11, GCR16, KJC+21], we want to introduce precision-oriented metrics specifically developed for evaluating information retrieval systems.

In information retrieval, there is a trade-off between precision and recall of a retrieval system: The trade-off between precision and recall arises because increasing precision leads to a decrease in recall and vice versa. This trade-off is primarily driven by two key factors.

The first key factor is thresholding. The retrieval system uses a ranking mechanism or a threshold to decide, which documents to present to the user. By setting a higher threshold, the system becomes more conservative and returns fewer results, which are more likely to be relevant, thus increasing precision but reducing recall. Conversely, lowering the threshold results in more retrieved documents, potentially improving recall but decreasing precision.

The second key factor is relevance ranking. The ranking of documents is heavily influenced by the retrieval model and the scoring functions employed. Some retrieval models focus on

optimizing precision by assigning higher scores to the documents that are more likely to be relevant. Others may prioritize recall by casting a wider net to capture a larger pool of potential relevant documents. These choices can have a significant impact on the precision-recall trade-off.

Thus this trade-off plays a critical role in the design and evaluation of retrieval systems, as it directly influences the performance and utility of retrieval systems.

Having discussed the trade-off between precision and recall, we want to delve into precision-oriented metrics other than precision. Precision-oriented evaluation metrics in information retrieval are essential tools for assessing how well a system retrieves and ranks documents, emphasizing the relevance and precision of the results. These metrics offer valuable insights into the quality of retrieved information by considering not only the number of relevant documents retrieved but also their order in the ranked list.. Three prominent precision-oriented metrics in Information Retrieval are Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG) [JK17] and Mean Reciprocal Rank [Cra09].

Mean Average Precision (MAP)

MAP [MRS08] is a widely recognized and highly regarded metric for assessing the performance of information retrieval systems [JK17]. It focuses on the precision and rank of relevant documents in the retrieved list. The key components of MAP are:

- Precision: For each query, precision is calculated by dividing the number of relevant documents retrieved by the total number of documents retrieved.
- Average Precision (AP): AP is computed by taking the average of precision values at various cut-off points in the ranked list. It rewards systems that place relevant documents at the top of the list.
- Mean Average Precision: MAP calculates the average AP across all queries, providing an overall measure of system performance. A higher MAP indicates better performance, with a maximum value of 1 when all relevant documents are ranked at the top.

Normalized Discounted Cumulative Gain (nDCG)

nDCG [JK02], like MAP, assesses the quality of ranked lists in information retrieval, with a specific focus on ranking relevance [JK17]. nDCG takes into account both the relevance of documents and their position in the list. Key aspects of nDCG include:

- Discounted Cumulative Gain (DCG): DCG assigns higher scores to relevant documents that appear at the top of the list, while gradually discounting the importance of documents lower in the ranking.
- Ideal DCG (IDCG): IDCG represents the best possible DCG score achievable for a given set of queries, where all relevant documents are perfectly ranked at the top.
- Normalized DCG (nDCG): nDCG is obtained by dividing the DCG by the IDCG. This normalization ensures that the metric has a value between 0 and 1, making it comparable across different queries and systems.

nDCG is aligned with the user's perspective, as it rewards systems for placing relevant items at the top of the list, which is crucial for user satisfaction and engagement [JK17]. nDCG can be calculated at various rank cutoffs (k), enabling a nuanced analysis of system performance. Different applications may prioritize different portions of the ranked list, and nDCG can accommodate these preferences.

Mean Reciprocal Rank (MRR)

Mean Reciprocal Rank (MRR) [Cra09] is a fundamental metric in the field of information retrieval and search engine evaluation. It focuses on the effectiveness of ranked retrieval systems by considering the rank of the first relevant document for each query. MRR is particularly valuable in scenarios, where the emphasis is on quickly providing the most relevant result to users.

The core idea behind MRR is straightforward: it calculates the reciprocal of the rank of the first relevant document for each query and then computes the average of these reciprocals across all queries in the dataset that is evaluated. In essence, MRR answers the following question: "On average, how quickly does the system return the most relevant result to users?"

The steps involved in calculating MRR are as follows:

- For each query, the system ranks the relevant documents based on their relevance to the query.
- MRR considers the reciprocal of the rank of the highest-ranked relevant document for each query. If a relevant document is ranked first, its reciprocal is 1; if it's ranked second, its reciprocal is 1/2; and so on.
- Finally, the MRR is computed as the average of these reciprocals across all queries.

MRR is particularly well-suited for evaluating information retrieval systems when users are primarily interested in quickly finding the most relevant information. It places a strong emphasis on effectiveness of retrieving that first relevant result. The higher the MRR score, the more effective the system is at promptly delivering valuable content to users, which is especially valuable in applications like web search and question-answering systems.

MAP, nDCG and MRR are highly valuable in information retrieval, especially for tasks like web search, document retrieval, and recommendation systems. They provide a nuanced assessment of a system's ability to retrieve relevant information, giving credit to the precision and rank of relevant documents in the final results. These metrics are pivotal in guiding the development and optimization of information retrieval systems to ensure that users receive high-quality and relevant information.

2.3 Domains

After introducing the evaluation metrics, we give a short introduction to the domains, which we consider in the thesis, and their characteristics.

2.3.1 Web Domain

Retrieval in the web domain refers to retrieval from the vast corpus of online content that can be searched, indexed, and retrieved [BP98]. This includes websites, blogs, forums, social media platforms, and other types of digital content that are accessible through the world wide web. The domain of web search is constantly evolving and expanding, with new content being added to the web every day [YHT+16]. Search in the web domain is thus characterized by its large volume.

Search engines like Google, Bing, and Yahoo are perhaps the most well-known applications of web information retrieval. These systems use algorithms to index and rank web pages, enabling users to find information based on their search queries. Techniques such as crawling, indexing, and ranking play a pivotal role in the operation of search engines [MRS08]. For web search engines it is crucial to be able to search the ever expanding content of the world wide web. Thus it is crucial for web search engine providers to have efficient and effective web crawling algorithms that automatically and systematically traverse the web in order to discover and collect web pages. Search engine crawlers or web spiders explore websites, follow links, and store data in a structured manner [HN99]. Efficient crawling is crucial for keeping search engine indexes up-to-date. Then the crawled web pages are stored in a distributed index to facilitate fast and relevant retrieval [BDH03]. These indexes typically store information about the content, keywords, and links on each page. Inverted indexes are commonly used to map terms to the documents, where they occur [RZ09].

In web search, the queries, which are issued by the user, are characterized by a short length and also the passages or documents in the collection are rather short compared to search in professional domains [NRS+16, SWJS01]. The queries include key word queries as well as questions [NRS+16].

When a user submits a query, web search engines employ query processing techniques to identify relevant documents. These query processing techniques include query parsing, where the query is parsed into individual terms [FSMZ10], query expansion, where the original query is expanded by some terms [LNP+18, ZCZ+23], or query term weighting, where the terms in the query are weighted based on their importance in the query [SLK+23]. To lower the users cognitive load when searching, web search engines contain a query auto-completion module, which suggests one or multiple completions to complete the query, that the user is typing at the moment [CdR16, HMRS14].

The users in web search are lay persons as well as domain experts with different information needs. Broder et al. [Bro02] identify and categorize the different information needs of users in web search to navigational, informational and transactional information needs. These needs induce various retrieval tasks. For a navigational information need, the intent of the user is to reach a certain web page, also referred to as known-item search. For the informational need, the

user wants to acquire some information that can be presented by multiple web pages. Broder et al. define a transactional need to reach a certain web page where further interaction takes place, like shopping or downloading files.

Relevance ranking is at the core of web search. In commercial web search engines relevance ranking includes different ranking functions, semantic matching features, query rewriting [YHT+16], incorporating implicit user feedback [ABD06, RKJ08], link analysis [PBMW98], anchor-text [CHR01] and features like recency [DCZ+10]. Furthermore for relevance ranking, commercial web search engines, like Google, also include features of ad revenue in their ranking models, optimizing the trade-off between relevance and ad revenue [RBC+08].

The advent of mobile devices has revolutionized the way people access information on the World Wide Web. Mobile devices encompass a range of devices, including mobile phones, smartphones, Personal Digital Assistants (PDAs), smartwatches, and other internet-connected devices that lack standard-sized screens or traditional keyboards [CMS19]. With the proliferation of smartphones and tablets, web search on mobile devices has unique characteristics and considerations that set it apart from traditional desktop web search [CMS19]. These changing characteristics include mobile-friendly design, voice search, location-based search, integration of apps and vertical search with various verticals, such as image search, video search, and news search.

Recently, personal voice assistants like Google assistant, Amazon Alexa or Cortana have emerged as personal assistants, where the users interact with the assistant via voice. Furthermore large language models like ChatGPT [Ope23] have emerged, where the user interacts with the language model in a conversational way. These novel technologies change, how users interact with systems and search for information, which is in a more conversational way. Thus the field of conversational information seeking [ZTDR23, DXC20] has emerged in the web domain, which is concerned with a sequence of interactions between one or more users and an information system and includes applications like conversational search [JOH+19], conversational question answering [ZZS+22], and conversational recommendation [LKS+18].

In the information retrieval community, various evaluation campaigns for web search were conducted over the course of more than 20 years in order to evaluate ranking models [HVCB99, CH02, CMY+21b, DXC20]. These many campaigns reflect the large interest and importance of web search in the information retrieval community.

2.3.2 Legal Domain

In the legal domain, lawyers, legal professionals and paralegals conduct searches in order to find prior cases to a given case, in order to analyze current regulations and statutes or to find evidence for a current case. Although there are also private persons conducting legal searches, we focus here on professional legal search conducted by legal professionals.

Search Example ❶ is also an example of a search process in the legal domain.

Where most of the legal searches were originally based on printed materials [How95], since the 1970s the electronically stored information in legal research has been exponentially growing [vOS17], so that nowadays most of the legal searches are conducted with online libraries. The

growing amount of electronically stored legal information also holds the risk that it is hard to distinguish the quality and impact of different court decisions of same legal status [BT07, MS08], therefore a legal professional needs to take the legal hierarchy of a document at hand into account.

Apart from the challenges of quantities of information in legal information search, also the quality of legal search is complex [vOS17]. Legal work is an intertwined combination of research, drafting, negotiation, counselling, managing and argumentation [LPS96]. Therefore the tasks in the legal domain are not limited to finding prior court decisions, which are relevant to a given case, but are a complex process of obtaining legal evidence. In the literature legal information seeking is defined as the behaviour displayed by lawyers when using a range of existing legal resources to find information required for their work [vOS17].

Legal documents are written in legal language [Glo23]. This legal language are characterized the use of precise, formal, and complex language in legal documents that establishes a framework for interpreting and enforcing the law while maintaining consistency and objectivity within the legal system [Tie00]. Legal language is precise and specific and aims to eliminate ambiguity and vagueness by using well-defined terminology and terms that have specific legal meanings [Tie00]. This precision is crucial to ensure that laws and legal documents are interpreted and applied consistently. Legal language is highly formal and often uses archaic or specialized vocabulary and also uses latin phrases, such as "pro bono", which are often used as legal shorthand for specific legal concepts [Tie00]. Furthermore legal documents have long, complex sentences and convoluted syntax. This makes legal documents written with legal language different to other documents in other domains and poses different challenges for retrieval of these documents.

Search in the legal domain has certain characteristics, which sets legal retrieval tasks apart from retrieval tasks in other domains. Turtle [How95] and van Opijnen et al. [vOS17] find specific characteristics of the legal domain that distinguish it from other domains. Some aspects relate to the legal texts itself, other with the way legal materials are used. The different characteristics are the following:

- Volume: Although the longstanding impressive volumes of legal materials are surpassed by web and social media data, the amount of legal data is still impressive [vOS17]. The US case law comprises roughly 50 GB of text which grows by 2 GB each year [How95].

- Document size: Legal documents have a longer average length compared to other domains like the web or social media domain. [RGK+22, BGG+19, AVA+23]

- Structure: Legal documents have a very specific internal structure, statutes and administrative codes have a hierarchical structure, case law documents have a jurisdiction specific structure [How95, PST07] containing for example summaries and the claims of the case [RKG+20, BGG+19]

- Heterogeneity of document types: there is a variety of document types ranging from legislation and court decisions to parliamentary documents, contracts, commentaries etc [vOS17]

- Self-contained documents: the documents in the legal domain are not just about the law, but they contain and constitute the law themselves [vOS17]
- Legal hierarchy: the legal documents are in a hierarchical organization with regard to the type of the documents and their authority. The importance of a document depends on its origin, for example a supreme court opinion overrules a municipal court decision. [Tur95, vOS17]
- Temporal aspects: legislation changes over time and therefore it is important to consider temporal aspects in the search [Tur95]
- Importance of citations: citations are an integral part of argumentation in the legal domain [vOS17, WV20]

These characteristics of the legal domain need to be considered when designing an information retrieval system for legal information seeking as well as when conducting a legal professional search.

2.3.3 Patent Domain

Innovation is crucial for the progress of technology and society. As technology builds on previous advancements, it benefits society when technical innovations are publicly available and well-described [LH13] and the patent system was established to achieve this goal by incentivizing inventors to share their expertise in exchange for temporary monopolies [SZ19]. According to the WIPO intellectual property handbook [wip04], a patent is a government-issued document that describes an invention and creates a legal environment, in which the patent holder is the only one authorized to exploit it. Patents provide immense economic value, and as the number of filed patent applications continually rises each year, there is an increasingly urgent need for effective systems to manage this massive amount of data [KJ19]. Retrieval in the patent domain aims to develop techniques and methods that can efficiently and effectively retrieve relevant patent documents in response to a given search request.

Patent documents are spread across various datasets, patent offices, and resources that require different patent search systems and online services, such as Google Patents and Espacenet, among others [Sal17]. Searching through multiple resources is crucial in certain patent search tasks with the goal of achieving the broadest coverage possible. As patent retrieval tasks are typically recall-oriented, it is essential to retrieve all related patent documents to avoid significant economic repercussions [KJ19, MGHC13]. Therefore, conducting an efficient and effective search across all patent sources is of utmost importance in the patent domain.

Search Example ❷ is an example for a search process in the patent domain, where the search process also demands a high recall.

2.3.4 Medical Domain

Scientific information in the medical domain is expanding rapidly, where a daily average of 75 clinical trials and 11 systematic reviews were published in 2010 alone [BGC10]. Similarly for

biomedical publications, more than 1 million papers are published on PubMed each year [Lan16], which is around 2 papers per minute. However, despite its growth in size, scientific information becomes outdated quickly, with new publications presenting recent experimental results that can alter the understanding of a topic or disprove previous findings. Additionally, researchers may prioritize maximizing their publication count, leading to fragmentation of literature and the emergence of specialized subfields of research [Her20]. The vast and evolving nature of the medical knowledge and thus the documents in the medical domain are one characteristic of the medical domain [XSW21].

Especially during the international Covid-2019 pandemic, the fast evolving nature of the medical domain was evident and the Covid-19 pandemic has influenced and challenged information retrieval in the medical domain [WLC+20, RAB+20]. The pandemic led to a massive increase in the demand for medical information and at the same time led to an unprecedented acceleration in the publication of medical research papers and studies about COVID-19 [RAB+20].

Another characteristic of the medical domain is its specialized terminology [AAZ18, CCHJ94, MC02]. The medical field relies on highly specialized terminology and jargon, including medical conditions, treatments, procedures, anatomical structures, and pharmaceuticals. The use of structured taxonomies like the MeSH taxonomoy [LB94, DGGCMV+08] is common in the medical domain to categorize and organize medical concepts and is also crucial to be integrated for retrieval.

Ensuring that the information retrieved is credible and trustworthy is also critical in healthcare [SR17, VP17, UPV21]. Retrieval systems should consider the source's reputation and the quality of the information.

It is important to recognize that different types of medical information require distinct retrieval approaches. For example, patient-specific information, which can be either structured or narrative, is most relevant to healthcare practitioners who work directly with patients. On the other hand, knowledge-based information such as experimental findings, summaries, and observations, is valuable to both clinicians and researchers as it can be applied to individual patient cases [Her20]. For sensitive patient data, there are strict regulations which govern the storage and retrieval of medical information. Also many healthcare providers use electronic health records systems and effective information retrieval should integrate with electronic health record systems to provide seamless access to patient information.

Retrieval tasks within the medical domain can be extremely time-sensitive, for example in cases where the Intensive Care Unit team must assess a patient's condition and electronic health records, and search medical literature to make critical decisions for treating the patient. Despite their best efforts to obtain the most accurate, up-to-date, and comprehensive information possible, these teams are often under tremendous pressure and have only a few hours, or less, to make their decision [Cas03].

Search Example ❸ is also an example of a search process in the medical domain that is extremely time-critical, as the doctor is working in the emergency department of the hospital and the treatment of patients requires quick decision making.

Other types of medical searches, such as literature searches for systematic reviews, are at the opposite end of the spectrum. These processes are slow, methodical, and involve multiple iterations and manual labor from qualified researchers. Recent studies suggest that a full systematic review can take an average of 67 weeks to complete [BBCK17].

There are some similarities between the search processes used in the medical domain and those applied to legal or patent applications. In both cases, conducting a formal search of secondary studies involves using detailed documentation and Boolean queries as standard procedures [RRM20, SZKC20, SZK21]. A Boolean query is a type of search query that allows users to combine multiple keywords or search terms using logical operators such as AND, OR, and NOT. Boolean queries enable users to create complex and precise search queries. While exact-match search tools are commonly used to retrieve all relevant documents [RZ09], it requires an understanding of operators and underlying databases, and thus tends to be aimed more towards "power users" [Her20]. However, ongoing research is being conducted on other approaches, often based on semi-automation, to form search queries when looking for information in electronic medical records [TTSS+20].

2.3.5 Conclusion

As introduced in this section, the web, legal, patent and medical domain have each different, special characteristics including the nature of information needs in each domain, the language of the queries and documents in each domain, the different structure of documents, the ever expanding volume of the collections, as well as temporal aspects. For designing and modelling an effective and efficient retrieval system it is crucial to take into account the respective characteristics of each of the domains and to know the domains, which tasks one wants to address and solve.

2.4 Tasks

The different characteristics of each domain affect the retrieval tasks, which appear in the domains [SW21]. A task is in general defined as a set of connected physical, cognitive, and affective actions through which individuals try to reach a goal [Bys07, MBB+07]. In the context of information retrieval, the task is the representation of the goal of the search process [SW21].

This section focuses on generic information seeking and retrieval tasks [MWL20] in the different domains. These have been classified by the intent such as search task versus a browse task. These are tasks that are “carried out by a [user] as a means to obtain information associated with fulfilling the work task” [IJ05]. Having introduced the specific domains, we now give a brief overview of some of the information seeking and retrieval tasks in each domain, which we identify as important and open challenges for retrieval systems and which are of particular interest for our work.

2.4.1 Tasks in the Web Domain

One retrieval and ranking task in the web domain is ad-hoc retrieval [HVCB99], which is a standard retrieval task in which the user indicates his/her information need with a query. This

query initiates a search for documents which are to be expected relevant to the user [BR11]. The task is called ad-hoc because the task is not planned nor part of a larger search process rather it is an information need that the user has in the moment. The queries can be keyword queries as well as questions [SWJS01].

Another task in web search is the web page finding task, which is defined as finding a specific web page [CH02]. This task refers to the navigational information need identified by Broder et al. [Bro02].

There are works studying information needs and tasks in a more fine-grained way in web search by analyzing the browsing behaviour and the goals of users [RL04, Dum13, BJ12]. Broder et al. [Bro02] categorize user behaviour in web search into navigational, informational and transactional information needs. Similarly Russell et al. [RTKJ09] identify different search tasks in web search including navigation, finding simple evidence (like looking up a phone number), finding complex information, which could be acquired from multiple sources, acquiring documents for example in order to download them, exploring/learning, and playing. Broder et al. as well as Russell et al. find that the information information need or informational tasks make up the most of the information needs/tasks in web search, compared to navigational and transactional information needs/tasks.

Bailey et al. [BJ12] further refine the tasks identified by Russell et al. by using the web search iterative taxonomy developed by Rose et al. [RL04]. They identify action-based tasks from user browsing behavior. Most frequent tasks include monitoring frequently updated information, browsing a social network, or comparing products or services for use. These tasks can but not have to be completed within one session and can expand over multiple search sessions.

With the rise of mobile devices for accessing the web the tasks in the web domain are changing and evolving and moving more to tasks like question answering [Cla18a, NRS+16], in order to fulfill the information need of the user. In question answering, the goal is not to find a document but an answer to a question thus the user does not have to browse through the web pages themselves. This can also be enhanced in the search engine result page by highlighting the relevant information of a web page [Hel23].

Recently, personal voice assistants like Google Assistant, Amazon Alexa, and Cortana have emerged as virtual companions, allowing users to interact with these assistants using their voices. Furthermore, the rise of extensive language models like ChatGPT [Ope23] has introduced a conversational way of interacting with information systems, transforming how users search for information. Consequently, the field of conversational information seeking [ZTDR23, DXC20], within the web domain, has come to the forefront. This field is concerned with the series of interactions between one or more users and an information system and includes applications like conversational search [JOH+19], conversational question answering [ZZS+22], and conversational recommendation [LKS+18].

The information retrieval community has conducted various evaluation campaigns for web search spanning over two decades, aimed at assessing ranking models [HVCB99, CH02, CMY+21b, DXC20]. These numerous campaigns reflect the substantial interest and significance of web searching within the information retrieval community.

2.4.2 Tasks in the Legal Domain

There are different types of retrieval in the legal domain ranging from traditional ad-hoc queries, where the user types in a query and expects to receive a set of relevant documents as response, to known-item search or navigational search to within document retrieval [How95]. We review the literature for search tasks in the legal domain and we describe two legal search tasks in more detail, eDiscovery and prior case retrieval.

The legal search tasks depends also on the different law systems, there are case law and statue law systems. In case law systems the body of law is created by judges and the precedent cases determine the law, and statute law systems, which is written law passed by a body of legislature. For the prior case/precedent retrieval task [AKTVJ01, Loc17, RKG+20, SML+20] in case law systems one only needs to return few top cases that have high conceptual relevance, possibly low keyword overlap and high juristic value for a query, concurring decision by lower courts can be safely ignored [vOS17].

In prior case retrieval typically a current case, called instant case [JAKTV03] is at hand for which relevant prior cases need to be retrieved. The task should lead to prior cases which should be taken into account for solving the current case [RKG+20], in other words which support or contradict the current case [SML+20]. The information source is primary literature containing previous court decisions and the queries are formed using keywords and Boolean operators [BT07, Tur94]. The desired output of the search is a list of prior cases, sorted by relevance or temporal aspects. In order to develop an effective defense strategy and to know all possible arguments from opponents, it is crucial, to find all related precedent legal cases, thus this task requires a high recall [RKG+20].

For example, **Search Example ❶** is a prior case retrieval task in the legal domain. In this example it is important to find all relevant prior cases, thus the retrieval model should aim and be evaluated for a high recall.

In statute retrieval [BGG+19] there is also a query case given and it is the task to retrieve the statutes which are relevant to solving the query case. In contrast to case law retrieval, in the statute retrieval task [BGG+19] the claim has to be complete and it would be highly critical for the legal process to miss a relevant statute.

Similar to prior case/precedent and statute retrieval there is also argument retrieval in the legal domain in order to support the argumentation line of legal practitioners [Ash14, AW13, MBPR07]. This task is useful to employ retrieved information in proposing new arguments, and to explain any proposed evidence or conclusions [AW13].

Another legal search task eDiscovery is widely represented and analyzed in the literature [BT07, Con10, GCHO11, OBH+10]. eDiscovery is defined as any process (or series of processes) in which electronic data is sought, located, secured, and searched with the intent of using it as evidence in a civil or criminal legal case. This data can include emails, images, audio or video files, calendar invitations, instant messages, spreadsheets, or computer programs. Court-ordered or government sanctioned inspection of data for the purpose of obtaining critical evidence is also a type of eDiscovery [Con10]. Here the lawyers' inquiry in discovery is intended to capture all or

as many as possible relevant materials of evidence to the case at hand [OBH+10], thus there is a emphasis on recall over precision. During the discovery phase of a litigation, one legal party can make a production request, which is a formal request, where one party asks the other party to produce or provide copies of specific documents, electronically stored information, or other types of evidence that are relevant to the case [GCHO11]. This production request can include an outline of specific documents or categories of documents that the requesting party seeks and is the initial information at hand to start the search task. The search is performed by a lawyer, paralegal or legal practitioner. The query is then formulated by extracting keywords and their synonyms from the production request and the formulation of a Boolean query [OBH+10]. The search results are then manually viewed to identify cases that are relevant to the production request.

2.4.3 Tasks in the Patent Domain

Patent search can occur at various stages in the patent lifecycle, performed by different stakeholders, and for a range of purposes. As a result, specific search tasks exist at different stages in the patent lifecycle with different information needs. These search tasks include prior art search, patentability, novelty, freedom to operate, and infringement [AYF+11]. However, a significant challenge arises from the lack of a common framework defining all the different search tasks that can be conducted. What is considered an identical task for one researcher may be regarded as a different category by another researcher. This variation is influenced by the level of detail considered when identifying tasks.

Azzopardi et al. [AVJ10] distinguished between novelty and patentability searches, while other researchers considered them identical. In their work, Azzopardi et al. considered novelty as the main search type, with patentability falling under the novelty category when a patent application exists. Through a pair-wise comparison, they found a high correlation between novelty and patentability. Shalaby and Zadrozny, in 2019, treated infringement search and freedom to operate search as different tasks [SZ19]. They defined infringement search as the main type of search with the goal of finding infringement, while freedom to operate extends beyond infringement search to give the freedom to sell products that do not infringe on existing patents. Additionally, patent landscaping was identified as the same as state of the art by Alberts et al. [AYF+11], with the only difference being in the way the results are presented.

Below is a summary of the different patent search tasks gathered from various works including [AYF+11, AVJ10, BCC10, Cla18b, HNR07, LH13, SZ19]

- state of the art/prior art search/patent landscaping: This task involves identifying the current state of the art in a particular field. It helps to identify potential competitors and opportunities for improvement. Since it is crucial not to miss patents that are prior art to the given patent, this task is recall-oriented. **Search example ❷** is an example for a prior art search task, since the patent attorney needs to identify the current state of the art, in order to file a patent application for the novel invention.

- novelty: This task involves determining whether an invention is novel and unique compared to existing patents and is an essential step in the patent application process.

- patentability: This task involves assessing whether an invention meets the legal requirements for obtaining a patent. It includes determining whether the invention is novel, non-obvious, and useful.
- validity: This task involves assessing the validity of an existing patent. It helps to determine whether the patent is enforceable and whether it can withstand legal challenges.
- infringement: This task involves determining whether a particular product or process infringes on an existing patent. Often this task is done if the product or process already exists [AYF+11]. It is important for businesses to avoid infringing on others' patents and for patent owners to protect their intellectual property.
- freedom to operate: This task is usually done when considering developing a technology or launching a new product or process to avoid potential legal issues and involves assessing whether the potential technology development or product launch infringes on existing patents.

Overall, these tasks vary in their scope, purpose, and methods, and they are essential for various stakeholders involved in the patent lifecycle, including inventors, businesses, and legal professionals.

2.4.4 Tasks in the Medical Domain

In the medical domain, there are various search tasks differing by their users, their timeframe and their output goal. In this section we will introduce ad-hoc retrieval in the medical domain and the task of systematic reviews.

Ad-hoc retrieval (health information seeking) in the medical domain is a process of searching for relevant information in a collection of medical documents or databases to answer a user's query for which the information need of the user occurs in the moment. In the medical domain, ad-hoc retrieval is often used to support clinical decision-making, medical research, and other healthcare-related tasks [MMM15]. Ad-hoc retrieval in the medical domain is challenging because of the complexity and diversity of medical knowledge, the use of specialized terminology, and the need for accurate and timely information. The users vary from domain experts like doctors to lay persons like patients [GKL14]. In this search task the focus is not be exhaustive and find all relevant documents, but to quickly find relevant documents. Thus this task is precision-oriented and aims at providing relevant information at a high rank of the result list.

For example, **Search Example ❷** is an example of an ad-hoc retrieval task in the medical domain. Here the doctor works in an emergency department of a hospital and needs to decide quickly on medical treatments of patients. He needs to have a search system at hand, that provides him with relevant evidence quickly. Thus the search system is required to have a high precision.

Systematic review is a search task with the goal of producing a secondary study that provides a comprehensive summary of all relevant data that meets pre-defined criteria to answer a specific research question. This approach employs rigorous scientific methods to minimize bias and derive robust conclusions, which can guide doctors or medical practitioners in their decision-making

process [JRJ+09]. The Cochrane Collaboration[1] has been producing systematic reviews of healthcare interventions globally since 1993. In order to verify all empirical evidence, researchers must find all publications relevant to the research question. Such publications are later on evaluated and interpreted. The initial information for the task are research questions and a pre-defined protocol which includes information on Population, Intervention, Comparison and Outcome (PICO). The protocol also includes specific study design requirements as well as inclusion and exclusion criteria [HTC+23]. The users conducting the search are information specialists or librarians. Each search query, if possible, is also peer-reviewed to check for errors that could reduce the recall [CBVC+18, HTC+23]. The search is conducted with multiple databases like CENTRAL, Embase or MEDLINE. The queries are Boolean queries extended with MeSH and thesaurus terms, in order to expand the search query with indexing terms, synonyms, abbreviations and spelling variations [JRJ+09, WLZ23]. The output of the search is a list of publications matching the Boolean query [SZKC20, SZK21]. The subsequent steps in the Systematic Review process are the selection, analysis and summarization of the list of relevant publications. Selection is conducted by at least two rounds that comprise first title and abstract screening followed by full-text document screening. Data from selected publications are later extracted and synthesised using quantitative methods (i.e. risk of bias assessment). The whole process is finally summarised in a Systematic Review report [JRJ+09].

2.4.5 Overview of Tasks and Conclusion

In order to give an overview of the different tasks in each of the domains, we visualize the different tasks in Table 2.1 including the task name, the domain that the task appears in, examples for that task, references and its characteristic.

In this section we have introduced various different tasks in each of the domains of web, legal, patent and health. Each of the tasks has different characteristics, different users (laypersons or professionals), different query and document characteristics and require a high precision and/or a high recall, in order to fulfill the information need of the user. When designing and evaluating an effective retrieval system for a specific task, it is crucial to take these special characteristics into account.

In this thesis, we propose and evaluate ranking and retrieval systems in the context domain-specific tasks, which take into account the special characteristics of the task. We investigate neural ranking and retrieval models for document-to-document retrieval task in the legal and patent domain, namely prior case retrieval and prior art search. Here we focus on a high recall in the evaluation of the neural ranking and retrieval models, since both tasks are recall-oriented. Furthermore we address the problem of limited evaluation data for the task of medical ad-hoc retrieval and address the problem of limited training data for the task of ad-hoc retrieval in the web and health domain. In our study, we aim to isolate the impact of active learning on the efficient annotation of training data, distinct from any potential influence coming from domain-specific neural ranking and retrieval architectures such as BERT-PLI or PARM. To achieve this, we explore active learning strategies within the context of "common" neural ranking and retrieval

[1] https://www.cochrane.org

Task	Domain	Users	Characteristic	References
Ad-hoc retrieval	Web	Layperson & Professionals	Precision-oriented	[HVCB99]
Web page finding	Web	Layperson	Precision-oriented	[CH02]
Question-Answering	Web	Layperson	Precision-oriented	[Cla18a, NRS+16]
Conversational Search	Web	Layperson	Precision-oriented	[ZTDR23, DXC20, JOH+19]
eDiscovery	Legal	Professionals	Recall-oriented	[BT07, Con10, GCHO11]
Prior case retrieval	Legal	Professionals	Recall-oriented	[JAKTV03, Loc17, RKG+20]
Argument retrieval	Legal	Professionals	Precision-oriented	[Ash14, AW13, MBPR07]
Prior Art Search	Patent	Professionals	Recall-oriented	[AYF+11, AVJ10, Cla18b, LH13]
Patentability	Patent	Professionals	Recall-oriented	[AYF+11, AVJ10, Cla18b, LH13]
Novelty	Patent	Professionals	Recall-oriented	[AYF+11, AVJ10, Cla18b, LH13]
Freedom to Operate	Patent	Professionals	Recall-oriented	[SZ19]
Validity	Patent	Professionals	Precision-oriented	[AYF+11, AVJ10]
Infringement	Patent	Professionals	Recall-oriented	[SZ19]
Ad-hoc retrieval	Medical	Layperson & Professionals	Precision-oriented	[GKL14]
Systematic review	Medical	Professionals	Recall-oriented	[JRJ+09, CBVC+18, HTC+23]

Table 2.1: Overview of the different search tasks that we introduced per domain and their characteristics in terms of focus on precision and/or recall.

model architectures [NC19, KOM+20, KZ20], which are primarily designed for handling shorter documents and do not explicitly cater to long documents. As these neural architectures are proposed for the task of ad-hoc retrieval in the web domain [NC19, KOM+20, KZ20] and are designed to handle short queries and document, that do not exceed the input length of the BERT encoder [DCLT19], we focus on the tasks of web and medical ad-hoc retrieval for studying the impact of active learning methods on the annotation efficiency. Here we focus in our evaluation on precision-oriented metrics, since ad-hoc retrieval tasks require a high precision.

2.5 Datasets

After presenting the domains and some selected retrieval tasks in the respective domains, we introduce the datasets and test collections, which are employed in our work and which model the respective retrieval tasks.

2.5.1 Datasets for Tasks in the Web Domain

In our experiments, we focus on the task of ad-hoc retrieval in the web domain thus we use publicly available training and test collections for that task. In our work we conduct experiments on the large scale MS Marco passage collection [NRS+16]. MS Marco is based on around 1 million queries sampled from Bing's query logs and contains 8.8 million passages in its corpus. The passages are extracted from 3.5 million web documents which are retrieved by Bing. Its passage training set contains $503k$ human-labelled training samples of relevant documents and

its test set contains $12k$ test queries. For the training and test queries there is roughly only one document labelled as relevant on average.

Based on the MS Marco passage collection there are multiple test collections created during the TREC Deep Learning track [CMY+21b] and we use the test set from the TREC Deep Learning track 2019 [CMYC19] and 2020 [CMYC20], since these test sets were available at the time of conducting the experiments and demonstrate a reliable evaluation across various retrieval systems in the TREC track.

The Deep Learning track 2019 was studying ad hoc ranking in the web domain, comparing neural ranking and retrieval models in a large data regime. As a result of the evaluation campaign the TREC DL 2019 test collection was created employing the Cranfield evaluation paradigm and using assessors from NIST organized by TREC [CMYC19]. The reusable test set for the passage retrieval task consists of 43 queries containing $9k$ judgements in total and was pooled from 75 runs of the participating groups.

In the TREC 2020 Deep Learning track another reusable test collection was created also based on the MS Marco passage collection and with a similar methodology as in to previous year. The passage retrieval test collection TREC DL 2020 consists of 54 test queries with $11k$ judgements and has demonstrated to be a reliable and reusable test collection for benchmarking neural ranking and retrieval models.

2.5.2 Datasets for Tasks in the Legal Domain

In the legal domain we focus on the document-to-document retrieval task of prior case retrieval introduced in Section 2.4. We go in more detail about related work on document-to-document retrieval tasks in Section 3.1. It is a document-to-document retrieval task since the query is a (potentially long) document and the items in the corpus to be retrieved are also long documents. For prior case retrieval there are multiple training and test collections publicly available, in our work we use the COLIEE training and test collections from the COLIEE evaluation campaign in 2019 and 2020 and 2021 [RKG+20] as well as the Case Law test collection published by Locke et al. [LZ18].

COLIEE [RKG+20] is a competition for legal information extraction and retrieval which provides datasets for legal case retrieval and case entailment. The collections in COLIEE are based on cases from the Canadian case law system and are written in English. The cases are provided in a structured format and contain paragraph sections of the legal cases. The COLIEE datasets are the respective datasets that were provided in the respective year of the respective challenge.

Task 1 of COLIEE is a document retrieval task, called the legal case retrieval task, where a query case is given together with a corpus of candidate documents. We refer to the COLIEE dataset as the dataset that was provided to the participants during that challenge. In COLIEE 2019 and 2020 [RKG+20] Task 1 was a re-ranking task, where for each query case a set of already retrieved cases was given and the task was to re-rank the given documents. Thus the COLIEE 2019 and 2020 datasets contain query cases and a list of already retrieved cases to be re-ranked. From 2021 onwards Task 1 was changed to a full retrieval task, where the whole corpus of 4415

legal cases was given along with a training set and test set consisting of query cases and their relevance judgements. The relevant cases are the cases which are referenced in the query case, the references are not on a paragraph-level but on the document-level.

In Task 1 of COLIEE 2021 [RGK+22], the legal case retrieval task, query cases with their relevance judgements on the document-level are provided together with a corpus of candidate documents. The corpus consists of 4415 legal cases and a set of training and test queries with relevance annotations is given. The training set continuously increased in size over the years resulting in 650 query cases with on average 5 relevant prior cases for COLIEE 2021 and the test set consists of 250 query cases.

Task 2 of COLIEE 2020 [RKG+20] involves the identification of a paragraph which entails the given query paragraph, called legal case entailment. When a paragraph of a legal case entails the query paragraph, it means that the legal reasoning, principles, or rules established in the paragraph of the found legal case are applied and extended to the query paragraph. We will refer to the dataset, which was provided in the Task 2 of COLIEE 2020, as COLIEE 2020 Task 2. This dataset contains relevance labels on the legal case paragraph level, given a query claim, a set of claims, which are candidates to be the entailing paragraph to the query claim, as well as relevance labels for the candidate claims. In COLIEE 2021, Task 2 [RGK+22] consisted of a training and testing sets containing 326 and 100 base cases respectively. The training data consists of a query, a noticed case and the paragraph number of the paragraph in the noticed case which entails the query paragraph, thus is relevant.

For a broader evaluation, we also conduct evaluation on the prior case retrieval test collection CaseLaw [LZ18]. CaseLaw contains a corpus of legal cases downloaded from CourtListener, providing cases with its text and additional data like the date on which the case was filed and a list of unique IDs of other cases that cite the particular case. The collection contains a corpus of $63k$ legal cases and 12 topics with 100 human-labelled query cases with 2600 assessments in total. The relevance assessments were manually conducted by two lawyers and one paralegal, who were familiar with the case law search task and did case law retrieval on a weekly basis in their jobs. The relevance assessments denote on average 7 relevant cases per query case.

2.5.3 Datasets for Tasks in the Patent Domain

Since we focus on document-to-document retrieval tasks in our work, we are particularly interested in the document-to-document retrieval task of prior art search in the patent domain. For conducting experiments on this task we employ the publicly available CLEF-IP collection [PH19, PLH13], which was created over the course of multiple the CLEF evaluation campaigns between 2009 and 2013.

The CLEF-IP collection provides a corpus of 3.5 million patents extracted from the larger MAREC dataset[2] which contains documents representing over 19 million patents published at the EPO (European Patent Office), USPTO (United States Patent and Trademark Office), WIPO (World Intellectual Property Office) and JPO (Japan Patent Office).

[2]The MAtrixware REsearch Collection `http://ifs.tuwien.ac.at/imp/marec`

There are two datasets of CLEF-IP of interest for our work: the dataset of the document-to-document retrieval task of prior art search and the dataset from the passage retrieval task starting from claims. Both datasets contain English, French and German queries, where we only consider the English queries and candidates. The relevance labels are based on the "X" citations which are manually assigned during the patent granting process by trained patent examiners. These citations are publicly available for all patents and thus can be extracted and used as a measure of relevance.

The task in prior art search is given a patent document to find relevant prior patents in the corpus. For this task there are 351 training documents available and 100 test queries with on average 3 relevant patents. The patent documents have a pre-defined structure and consist of a title, abstract, the legal claims and a technical description. Often the technical description also contains technical drawings and pictures, however in these tasks only the text of the patents is taken into account [PLHZ11].

The task in the passage-level retrieval task is given a set of claims occuring in a patent application to find documents from the corpus which are considered relevant to the claims and to identify the passages of the documents which are relevant to the claims. The training set consists of 44 claims and the test set of 42 claims. The patent documents here have the same structure as in the prior art search task.

2.5.4 Datasets for Tasks in the Medical Domain

In the medical domain, we focus on the task of ad-hoc retrieval (health information seeking) and employ in our work the recently proposed benchmark of the publicly available TripClick collection [RLS$^+$21]. We choose this dataset because it contains a large-scale training and test set, different to other datasets in the medical domain, where only a small training dataset [RIS$^+$20] or no training dataset [RDV$^+$19, RDV$^+$17] is available.

TripClick contains large-scale real user queries and real user click logs from Trip, an English health search engine with professional and non-professional users. The usage of the Trip search engine is for free and thus attracts expert as well as non-expert users. The TripClick dataset contains real user queries and click-based annotations as well as a collection of medical articles and documents. The collection contains 1.5 million passages from MedLine articles, consisting of the title and abstract of the article. TripClick contains $680k$ training queries with an average length of 6 words. The queries are mainly keyword queries or short questions. Test queries are divided with respect to their frequency into three sets of $1,750$ queries respectively; the three sets are Head, Torso, and Tail. For the Head queries a Document-Click-Through-Rate (DCTR) model [CMdR15] was used to create relevance signals from the click labels. This results in multiple test sets with labels based on the clicks of the users, either estimating relevance by the raw clicks ('Raw') or by the rate of clicks of a document over all retrieved documents for a query ('DCTR') [CMdR15]. This dataset is representative of the task of health information seeking since it contains real user queries and real user interactions from the real, live health search engine Trip.

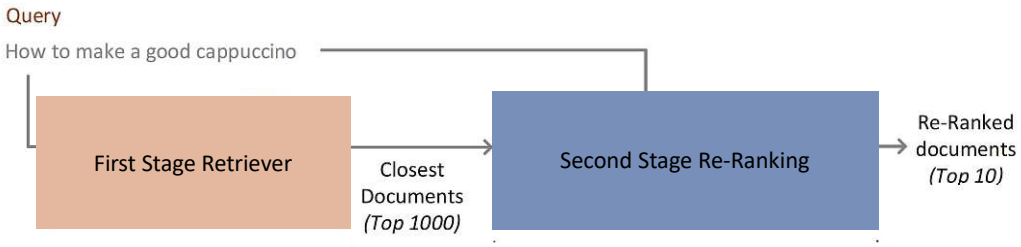


Figure 2.1: Retrieval Workflow with first stage retrieval and second stage re-ranking

2.6 Models

We visualize the general retrieval workflow in Figure 2.1. The general retrieval workflow consists of a query that the system receives, in our example the query is "How to make a good cappucino". Then the system employs a first stage retrieval model and retrieves the most relevant documents from a large corpus. How many documents are retrieved is a variable that can be chosen in the retrieval process, usually one uses the top 1000 documents. For the first stage retrieval process it is important that the retrieval model is very efficient, since it needs to score (potentially) billions of documents [NRS+16] in the whole corpus. At the same time the first stage retrieval model is required to have a high recall, since we do not want to miss any relevant documents from the collection.

In the second stage a re-ranking model is employed that re-ranks the top documents, that were retrieved in the first stage. Then the re-ranked model yields a re-ranked list of documents, depending on the task it can crop this list to only top 10 or display the whole re-ranked top 1000 list. For the second stage re-ranking model it is important that it has a high precision, in order to rank relevant documents highly. Furthermore second stage re-ranking models can be computationally more heavy, since the re-ranking model only needs to score the top documents, that were retrieved in the first stage.

In the following we will introduce the ranking and retrieval models that we employ and expand on in this thesis. We first introduce the statistical ranking and retrieval model BM25 and then describe the architecture and training of neural re-ranking and retrieval models.

2.6.1 Statistical Models

There are various different statistical retrieval models. Since we use the statistical ranking and/or retrieval models as baseline for comparing their performance to neural ranking and retrieval models, we choose the one statistical model as baseline, which has a robust, high retrieval and ranking performance across various retrieval and ranking tasks, called BM25 [RZ09].

BM25 is a ranking function that evaluates a collection of documents based on the query terms present in each document, regardless of the relationship between these terms within a document, such as their proximity. It is based on the probabilistic retrieval framework developed by Robertson et al. [RZ09]. BM25 is not a single function, but rather a collection of scoring

functions that differ in their components and parameters. It is a member of the BM family of retrieval models, which stands for Best Match. The most prominent instance of the BM family is BM25, which scores relevance as follows: The score of a query q, containing the key words $q_1, .., q_n$ and a document d is:

$$s = \sum_{i=1}^{n} IDF(q_i) \frac{TF(q_i, d)(k_1 + 1)}{TF(q_i, d) + k_1 \left(1 - b + b \frac{|d|}{avgdl}\right)}$$

with the term frequency function $TF(q_i, d)$, the inverse document frequency function $IDF(q_i)$ and $avgdl$ the average document length in the collection. b and k_1 are hyperparameters to be optimized for each retrieval collection.

The term frequency $TF(q_i, d)$ is defined as the number of occurrences of a query term q_i in document d. The inverse document frequency of a query term q_i is defined as:

$$IDF(q_i) = ln \left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1 \right)$$

where N is the number of documents in the collection and $n(q_i)$ is the number of documents containing q_i.

2.6.2 Neural Re-Ranking Models

We introduce two neural re-ranking models, namely cross-encoder BERT (MonoBERT) and ColBERT, both which we employ in our research. In this thesis we also refer to neural re-ranking models as simply neural ranking models.

Neural models are divided into re-ranking and retrieval models with different computational complexity. Theoretically neural ranking models could also be used to score the whole collection and retrieve relevant documents, but due to the models' high computational complexity and thus the models' slow latency at query time, the neural re-ranking models are used to re-rank a list of top N documents. This list of top N documents is retrieved in the first stage retrieval by either a statistical or highly efficient neural retrieval model.

The neural ranking and retrieval models use transformer-based [VSP+17] pre-trained language models like BERT [DCLT19] as backbone to encode the text. There are different pre-trained language models with the same architecture as BERT, and depending on the domain of the text which we want to encode we choose the backbone language model. For example we employ the SciBERT model [BLC19], which is pre-trained on scientific texts, or the PubMedBERT model [GTC+21], which is pre-trained on PubMed articles and medical literature, for encoding medical texts. Similarly we use the LegalBERT [CFM+20] model, trained on European, UK and US legislation texts, for encoding legal texts.

In order to decrease the computational complexity at training and inference time, one can also employ smaller BERT-like models with fewer layers. In our work we use DistilBERT [SDCW19]

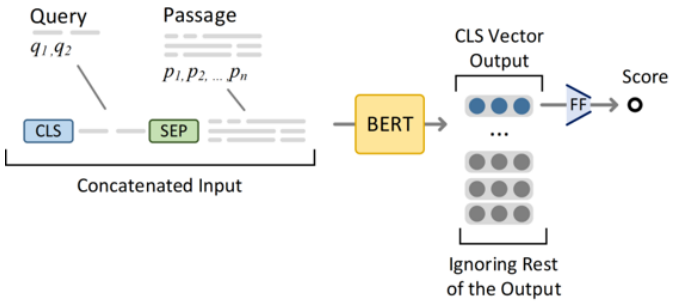


Figure 2.2: Architectural diagram of a MonoBERT model, taken from TU Wien, Advanced Information Retrieval lecture `https://github.com/sebastian-hofstaetter/teaching/tree/master/advanced-information-retrieval`.

as efficient encoding model, which is trained with knowledge distillation from the original BERT model and has 40% less parameters compared to BERT.

The BERT models convert the text input to tokens. The tokens can be whole words or subwords [ZFM+19]. This process is called tokenization. Each token is then linked to an embedding representation and this representation is learned during the pre-training. In its default architecture, a BERT model has an input length of 512 tokens and is thus limited in the amount of text it can process at once.

Cross-encoder BERT (MonoBERT)

In the cross-encoder model, MonoBERT, query and passage text are concatenated, encoded with BERT [DCLT19] and the CLS representation from BERT is scored with a linear layer W on top of the encoding:

$$s = W \text{ BERT}(\text{CLS}; q; \text{SEP}; p; \text{SEP})_{CLS} \tag{2.1}$$

where SEP is the separator token and s is the final score of passage p for query q. In Figure 2.2 there is also an architectural diagram of the MonoBERT model, where the concatenation process of the query and passage, the encoding with BERT and prediction of relevance score is visualized.

Empirical findings show MonoBERT reaches a high re-ranking effectiveness [NC19], however each passage needs to be encoded at query time and therefore this architecture is computationally resource-heavy and is characterized by high query latency [SZZ22, HH19]. For the same reason, this ranker is commonly used only in top-N re-ranking settings, and not for retrieval (i.e., scoring the whole collection for each query) [NC19, NYCL19].

MonoBERT is trained using training triples consisting of the query text, the text of a relevant passage/document (also denoted as positive passage/document) and the text of an irrelevant passage/document (also denoted as negative passage/document). During training different losses can be employed, in our implementation we use the RankNet loss [Bur10], if not noted otherwise. RankNet loss maximizes the difference between the relevance score of the positive and the negative passage using the Binary Cross Entropy loss as implemented in PyTorch [3].

[3] Binary Cross Entropy loss with logits used from PyTorch

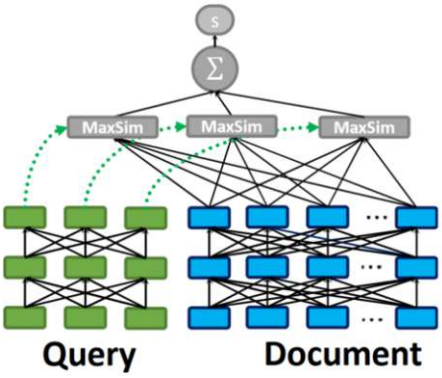


Figure 2.3: Architectural diagram of a ColBERT model, taken from [KZ20].

ColBERT

The ColBERT method delays the interaction between the query and passage to after the encoding by computing the relevance score as the sum of the maximum similarity scores between all token representations of the query and passage:

$$s = \sum_j \max_i \left[\text{BERT}(\text{CLS}; q; \text{SEP})_j \cdot \text{BERT}(\text{CLS}; p; \text{SEP})_i\right] \tag{2.2}$$

As for single representation bi-encoder methods, also in ColBERT the passage representation can be pre-computed offline and thus the query processing is sped up. In Figure 2.3 the ColBERT is visualized as architectural diagram, where the independent encoding and the maximum similarity relevance score computation is visualized. Empirical results show that ColBERT achieves a competitive effectiveness compared to MonoBERT [KZ20] while minimizing query latency compared to MonoBERT.

Like MonoBERT, ColBERT is also trained with the training triples and the RankNet loss, if not noted otherwise.

2.6.3 Neural Retrieval Models

The dense passage retrieval model (DPR/bi-encoder) [KOM+20] encodes the query and passages independently using a language encoder model like BERT [DCLT19]. The relevance of a passage p to a query q is estimated using the dot-product between the CLS token representation q and that of p:

$$s = \text{BERT}(\text{CLS}; q; \text{SEP})_{CLS} \cdot \text{BERT}(\text{CLS}; p; \text{SEP})_{CLS} \tag{2.3}$$

The independence of query and passage encoding and dot-product relevance scoring make it possible to pre-compute and store the passage representations in the index and enable efficient retrieval at query time with approximate nearest neighbor search [MY20, JDJ19]. In Figure 2.4 a dense passage retrieval model is visualized with an architectural diagram. The diagram visualizes the independent encoding of query and passage and the relevance scoring with the dot-product between their respective representations.

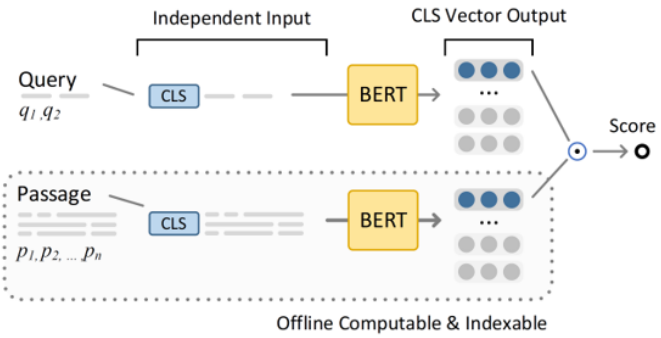


Figure 2.4: Architectural diagram of a dense passage retrieval (DPR) model, taken from TU Wien, Advanced Information Retrieval lecture `https://github.com/sebastian-hofstaetter/teaching/tree/master/advanced-information-retrieval`.

DPR is trained on triples of query text, the text of a relevant and an irrelevant document. For training the DPR model effectively Karpukhin et al. [KOM+20] propose in-batch negatives. In-batch negatives are a method to increase the number of training triples without adding additional computational complexity during training. In the training implementation, the query text and the positive text are also paired with the negative texts from the triples in the same batch and the loss is also computed for the triples with the in-batch negatives and added to the loss of the original training triples. This study is extended on by Qu et al. [QDL+21]. They optimize the training of the DPR model by selecting hard negative samples in the triples. The idea behind selecting hard negatives is that it is harder for the retrieval model to distinguish the positive passage from a hard negative and thus training with hard negatives results in higher retrieval quality [QDL+21, ZML+21].

In this thesis, we follow the implementation of Karpukhin et al. [KOM+20] for in-batch negatives. For selecting negatives for the training triple, we follow Karpukhin et al. [KOM+20] and select negatives from the BM25 top 1000 documents, which are not labelled as relevant in the training set.

CHAPTER 3

Related Work

In this section we review the field and set the scene by outlining the related work on document-to-document retrieval tasks as well as on the lack of data for evaluation and training of ranking and retrieval models.

3.1 Document-to-Document Retrieval

In our work we focus on document-to-document retrieval tasks, thus we will give a brief definition of document-to-document retrieval and will lay out related work for document-to-document retrieval tasks as well as other approaches for handling long documents for text processing.

3.1.1 Document-to-Document retrieval tasks

We define document-to-document retrieval tasks as those where the query is a (long) document and the items in the collection to be retrieved are also long documents. With long documents we refer to documents that greatly exceed the average length of queries in the web domain [HVCB99]. Document-to-document retrieval tasks are also referred to as query-by-example tasks [AVA22] or extremely long queries and documents [AVA$^+$23] tasks in the related literature.

When we recall **Search Example ❶**, where the attorney is retrieving similar cases to a given current case, we see that this example is a document-to-document retrieval task. Similarly **Search Example ❷**, where the patent attorney is searching related, granted patents to his new patent application, is also an example for a document-to-document retrieval task.

If the citations are used as label of relevance in a task, one could suggest that citation prediction [ABHH21] is the same task as document-to-document retrieval. Furthermore one could view document similarity prediction as document-to-document retrieval task since similar documents should be retrieved in document-to-document retrieval tasks. However we differentiate document-to-document retrieval tasks from document similarity or citation prediction, as the latter two tasks

are classification tasks and not retrieval tasks. In the web domain there are also various efforts addressing document retrieval [CMY+21b], however here the queries are short and thus this task has different characteristics than document-to-document retrieval.

Many documents in document-to-document retrieval tasks have a pre-defined structure like paragraphs or different sections of a document. For example legal cases in prior case retrieval are structured by their paragraphs or patents are structured in title, abstract, claims, technical description and possibly technical drawings and images.

There are many document-to-document retrieval tasks [CFB+20, MOMZ21, NFIH10], explored in the information retrieval community; we will focus in this work on the tasks of prior case retrieval in the legal domain and prior art search in the patent domain. These tasks are challenging retrieval tasks which are actively studied in the Information Retrieval community in various evaluation campaigns and competitions like CLEF [PLH13], COLIEE [RKG+20] and FIRE [BGG+19] and are tackled with a variety of different retrieval approaches. Having introduced the retrieval tasks and the related datasets in Section 2.4 and 2.5 for prior case retrieval and prior art search, we will now outline the different retrieval approaches for these document-to-document retrieval tasks.

The CLEF-IP evaluation campaign addresses the document-to-document retrieval task of prior art search [PH19] over the course of multiple campaigns form 2009 to 2013. Overall Piroi et al. [PH19] find that expertise in the Intellectual Property (IP) domain is essential for implementing Information Retrieval (IR) approaches to assist with certain tasks within this field requires domain-specific knowledge.

3.1.2 Approaches for Document-to-Document Retrieval Tasks

The retrieval approaches employ statistical and neural retrieval models. There are various strategies for handling long documents including key word extraction, summarization of the long documents and exploiting the inherent structure of the documents for retrieval. In Figure 3.1 we summarize the different approaches for handling long documents for document-to-document retrieval tasks or text processing in general and link to the relevant publications, that we describe in the following two chapters in more detail.

Within the FIRE AILA workshop for legal case retrieval, Gao et al. [GNS+19] handle the long query documents by extracting topic words from the given case and use the topic words as query. They investigate the vector space model [SWY75], the probabilistic BM25 model [RZ09] and a language model [SC99] to identify relevant legal prior cases with the topic word queries and reach competitive performance compared to other runs. In the same workshop, Zhao et al. [ZNL+19] reach the highest effectiveness by extracting the top 50% query terms from the query document with the highest inverse document frequency (IDF) and using the extracted terms as keywords for retrieval with BM25. In the following year also key word extraction approaches combined with BM25 dominated the leaderboard [LMTM20, LLH20].

For the COLIEE prior case retrieval dataset described in Section 2.5, statistical and neural ranking models show great effectiveness combined with key word extraction, summarization of long documents and chunking the documents in passages for handling the long queries and documents.

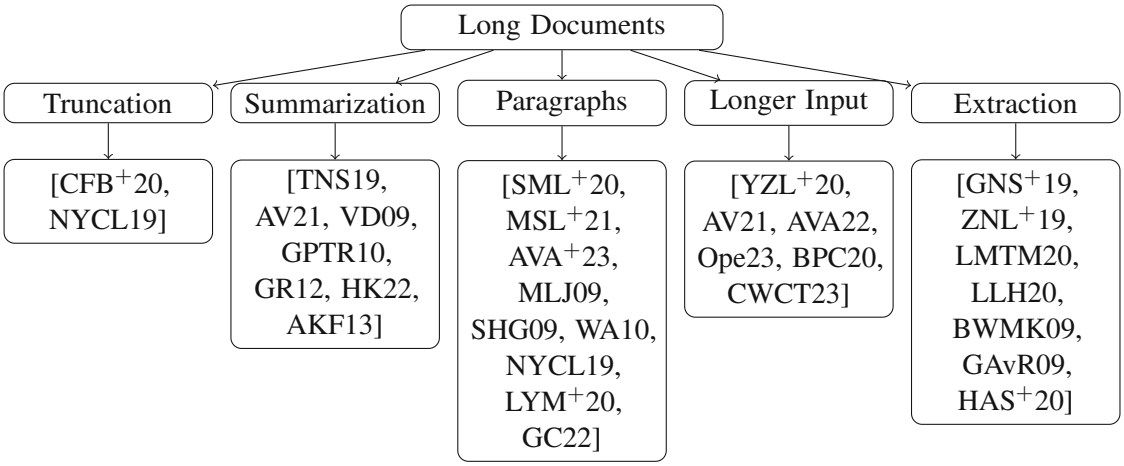


Figure 3.1: Approaches for handling long documents for document-to-document retrieval tasks or text processing

Shao et al. [SML+20] train a re-ranking MonoBERT model for scoring chunked passages of the query document and with the chunked passages of a candidate document, where the candidate documents are retrieved in the first stage with BM25. The scores of each passage are then aggregated to an overall score of relevance between the query document and the candidate document. Since the MonoBERT re-ranker has a limited input length of 256 tokens for the query and 256 tokens for the document, the approach of Shao et al. makes it feasible to take into account the whole query and candidate document for re-ranking. However this approach is highly computationally expensive, since one BERT inference is already costly and the number of BERT inferences for scoring one candidate document to a query scales quadratically with the number of passages of the query and candidate document. This approach is extended by Ma et al. [MSL+21] by adding a filter after the re-ranking model to remove unreasonable candidates from the result list. Askari et al. [AVA+23] extend the passage-level re-ranking by embedding chunked sentences with a dense retrieval model and re-ranking the documents based on a proportional relevance score.

In the COLIEE evaluation campaign, Rossi et al. [RK19] combine text summarization and a generalized language model to predict pairwise relevance for the legal case retrieval task, whereas Tran et al. [TNS19] apply a summarization method and the extraction of lexical features for handling the long documents. They rank the candidates using an early neural phrase scoring model and a learning-to-rank model. Ranking based on the summaries of the query and candidate document is also employed for a neural re-ranking model based on BERT by Askari et al. [AV21]. However optimizing the BM25 model with additional extracted key words from the query demonstrates higher effectiveness than the re-ranking with a trained MonoBERT model.

For handling the long queries and the long patent documents for retrieval, different approaches of either extracting query terms from the query patent or using the structure of the patent documents and splitting the documents into its different textual fields or in passages are employed for retrieval in the CLEF-IP evaluation campaign.

Since to the date of the CLEF-IP evaluation lab no neural, transformer-based ranking models had been developed yet, most participants experimented with statistical retrieval models extending them with machine learning classification models like k-nearest neighbour algorithms or support vector machines. These machine learning models were trained with the available training set. However the results with the higher effectiveness leveraged domain-specific knowledge and exploited patent specific data like citations or IPC classes [LR09, LR10, MJ10, MGHC13].

Participants studied which part of the patent document to take into account for ranking or retrieval and which contributed the most to improve the retrieval result. These approaches included experiments to select which text parts of the patent document to index and to weigh the scores of query terms extracted from certain parts of the patent documents [VD09, GPTR10, GR12]. Some of the participants decide to index the whole text of the document by concatenating all text fields, while some other approaches index the different parts of the patent document separately or construct indices on the passage-level of the patents [MLJ09, SHG09, WA10]. Several studies investigate which key word queries to extract from the query patent documents in order to achieve higher effectiveness than just retrieval with the whole patent document [BWMK09, GAvR09].

3.1.3 Handling long documents for text processing

After having introduced document-to-document retrieval tasks and approaches for handling long queries and long documents, we also want to outline general approaches how long document text is handled in the literature.

A natural but computationally complex approach to take into account longer text of the query and the document is to employ Transformer models which are designed to handle long input sequences like LongFormer [BPC20], however these approaches have not shown high effectiveness for document-to-document retrieval tasks [AV21]. However training MonoBERT with a multi-task ranking objective to train with longer input lengths shows competitive effectiveness for two document-to-document retrieval tasks [AVA22]. Liu et al. [YZL+20] propose similar document matching for documents up to a length of 2048, however here the input length is still bounded and the computational cost of training and using the model is increased. With the recent rise of language models with extremely long input length [Ope23, BPC20, CWCT23], it becomes an open direction if these large language models with long input lengths are more suitable for encoding long queries and documents for ranking and retrieval tasks.

Another approach for handling long text of document is generating summaries of the text and then using the summaries of the documents as textual representation. Askari et al. [AV21] train a LongFormer model [BPC20] to produce summaries of legal cases and then train a MonoBERT re-ranking model on the summaries of the legal text. While maintaining a high effectiveness by pre-computing the summaries of the legal cases, this re-ranking approach is outperformed by a lexical retrieval model, which hyperparameters are optimized for the retrieval task. Hartl et al. [HK22] also use summaries of news articles for fake news detection. They propose a CMTR-BERT framework that integrates various text representations to overcome the inherent sequential limitations and information loss of the off-the shelf transformer architecture. Alhindi

et al. [AKF13] use profile-based summarization to provide contextualization and interactive support for site search and enterprise search.

Another approach for handling long text is learning to select, which of the document parts are a relevant context for the underlying task. Hofstätter et al. [HAS+20] propose an intra-document cascading approach for re-ranking long documents to a given query, which is a passage-to-document retrieval task. This strategy involves first employing a more cost-effective model, referred to as ESM (Efficient Student Model), to filter out passages from a candidate document. Subsequently, a MonoBERT model, that is more computationally intensive and more, is utilized for re-ranking. Their optimization process involves training the ESM through knowledge distillation from the MonoBERT model. This distillation process enables the MonoBERT model to select and run only on a reduced set of passages, ensuring a consistent passage size across documents regardless of their length and also reducing the computational complexity of re-ranking the documents.

For the task of re-ranking long documents to a given query, the most simplistic approach is to truncate the text of the document, when the passage of the document exceeds the input length [NYCL19]. While this approach is easy to implement, the effectiveness of the approach depends on the characteristics of the document, namely if the relevant information of the document for ranking/retrieval is at the beginning of the document. This approach has the risk to miss information that is relevant for the specific task, if this information is contained later in the text of the document after the truncation limit. To mitigate this risk, Li et al. [LYM+20] propose to split up the document into passages and then train a MonoBERT re-ranking model on re-ranking each of the passages. Here simply the maximum passage score of a document is taken as the overall document relevance score. Thus with this approach the whole text of the document is considered for re-ranking. Similarly instead of modelling the full query-to-document interaction, Gao et al. [GC22] propose to leverage the attention operator and a modular Transformer re-ranking framework. In the first step, they independently encode individual document chunks through an encoder module. Subsequently, an interaction module encodes the query and facilitates joint attention, allowing for interaction between the query and all document chunk representations. They show the benefits of their novel approach, which offers the retrieval model the flexibility to aggregate relevant information from the entire document.

The passage level influence for retrieval of documents has been analyzed in multiple works [BK08, LC02, WML+20, WML+19] and shown to be beneficial, but in these works the focus lies on passage-to-document retrieval. Cohan et al. [CFB+20] present document-level representation learning strategies for ranking, however the input length remains bounded by 512 tokens and only title and abstract of the document are considered. Abolghasemi et al. [AVA22] present multi-task learning for document-to-document retrieval. Liu et al. [YZL+20] propose similar document matching for documents up to a length of 2048 however here the input length is still bounded and the computational cost of training and using the model is increased.

3.1.4 Aggregation strategies in Information Retrieval

In the context of breaking down document-to-document retrieval tasks at the passage level, it becomes necessary to explore methods for consolidating passage-level results into a cohesive document-level representation. Consequently, we will examine prior research on aggregation strategies within the field of Information Retrieval.

Aggregating results from different ranked lists has a long history in IR. Shaw et al. [Lee97, SF94] investigate the combination of multiple result lists by summing the scores. Different rank aggregation strategies like Condorcet [MA02] or Borda count [Wu12] are proposed, however it is demonstrated [CCB09, ZYL21] that reciprocal rank fusion outperforms them. Ai et al. [AOC18] propose a neural passage model for scoring passages for a passage-to-document retrieval task. Multiple works [AYWY+19, AYYZL19, DC19, ZYL21] propose score aggregation for re-ranking with BERT on a passage-to-document task ranging from taking the first passage of a document to the passage of the document with the highest score. Different to rank/score-based aggregation approaches, Li et al. [LYM+20] propose vector-based aggregation for re-ranking for a passage-to-document task. Different to our approach they concatenate query and passage and learn a representation for binary classification of the relevance score. The focus of score/rank aggregation is mainly on federated search or passage-to-document tasks, however we focus on document-to-document retrieval. We have not seen a generalization of aggregation strategies for the query and candidate paragraphs for document-to-document retrieval yet. Different to previous work, we propose to combine rank and vector-based aggregation methods for aggregating the representation of query and candidate documents independently.

3.1.5 Summary

The various evaluation campaigns for prior case retrieval and prior art search promote research in these fields and show the great interest in the Information Retrieval community in document-to-document retrieval tasks as well as the impact of improving the performance for these retrieval tasks for the stakeholders in professional search. Document-to-document retrieval tasks pose various interesting and widely discussed challenges for statistical and especially for transformer-based neural ranking and retrieval models, the models we focus on in this work. The length of the query document and the documents in the collection usually exceeds the input length of transformer-based neural ranking and retrieval models. Thus it is not possible to take the text of the whole query document and of the whole document, which is ranked or retrieved, into account for neural ranking and retrieval with off-the-shelf solutions. However it is crucial for a high ranking and retrieval effectiveness to not miss important parts of the query and candidate document. Related work demonstrates that ranking and retrieval with long documents as query and documents in the collection is not trivial and various different lines of research exist when it comes to the question how to handle the long documents in the best way. It is an open research question how to handle long documents in document-to-document retrieval tasks for neural ranking and retrieval models including aspects of efficiency of the ranking/retrieval process.

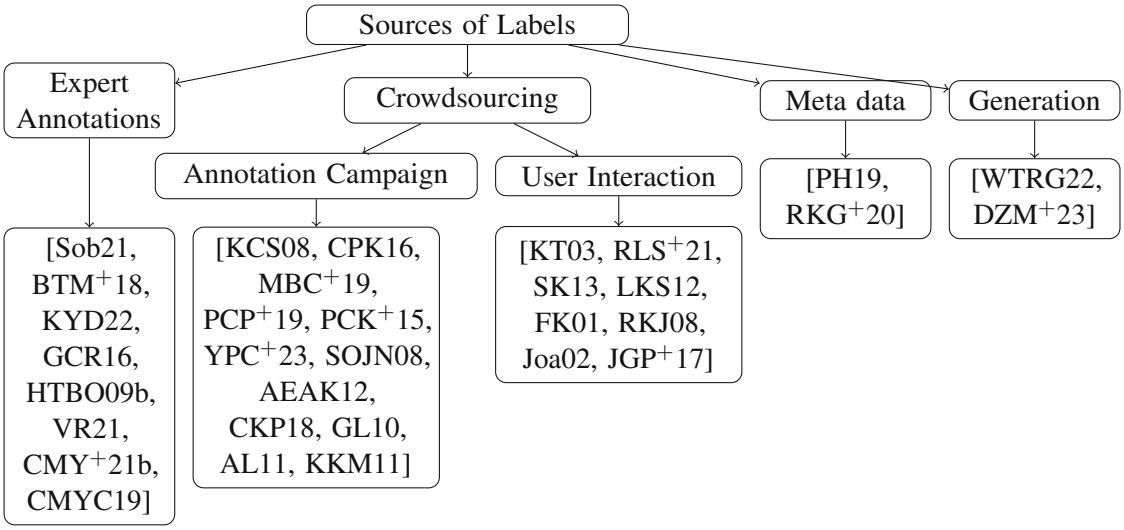


Figure 3.2: Sources for attaining labels

3.2 Lack of Data

The lack of data for evaluation and training poses a significant obstacle to the progress of research within the Information Retrieval community. In the following we lay out the research landscape for different strategies of attaining labels including evaluation campaigns with expert annotators, crowdsourcing with annotation campaigns or user interaction or using meta data as labels. We categorize the related work in Figure 3.2 and link to the relevant literature that is described in more detail in the following chapters.

3.2.1 Evaluation campaigns

Thus there is a long history of evaluation campaigns in IR like TREC [Sob21], CLEF [BTM+18] and NCTIR [KYD22]. The goal of these evaluation campaigns is promoting research for certain retrieval and ranking tasks by providing a forum to compare and measure different retrieval approaches of the participating teams during the campaign and by providing reusable and reliable evaluation sets after the campaign. Depending on the task also training sets for training and fine-tuning of ranking and retrieval models are created and made publicly available.

An evaluation set is a dataset that is used not to train a model, but to evaluate the performance of a model using metrics, that are defined for the retrieval task or the evaluation set. Evaluation sets need to fulfill certain requirements: they should be reliable and reusable. Thus a great effort in IR research is put into creating test collections that fulfill these requirements [Zob98, Voo18, VSL22, Sob17], especially in those task-specific evaluation campaigns. These campaigns follow the Cranfield paradigm [Cle91] to create relevance judgements on the pooled output of the participating systems as well as multiple instructed assessors. An IR test collection is reliable if the annotations of the samples in the evaluation set and the resulting metrics reflect the overall users preference between two systems. An IR test collection is called reusable if it unbiased

towards systems that did not participate in the collection-building process [VR21]. The most common way of testing the re-usability of a collection is the Leave-Out-Uniques (LOU) test [BDSV07, Zob98]. The LOU test compares rankings from two test collections, the original collection and a reduced collection in which the relevant documents that were contributed to the pools by a single run are removed. One rule of thumb to achieve re-usability of the test collections is to judge the pooled runs deeply, so that at least 66% of judged documents are irrelevant [Voo18, CMYC19]. Furthermore a large variety of retrieval approaches of the runs, which are participating to the pool, diversifies the pools and leads to a reusable test collection. Zobel et al. [Zob98] analyze the reliability of the TREC test collections and their empirical investigation demonstrates that the evaluation results based on the relevance assessments formed from a limited depth pool are reliable in case the pool is sufficiently deep for systems that contributed to the pool.

There are tracks, in order to evaluate models for certain retrieval tasks [GCR16, HTBO09b, VR21], and a track to specifically evaluate neural ranking and retrieval models when training data is available in a large amount [CMY+21b, CMYC19]. Although old TREC test collections did not have neural ranking or retrieval models as participating runs, since neural models did not exist back then, Voorhees et al. [VSL22] find that old TREC collections are still able to reliably evaluate neural ranking and retrieval models.

Creating test collections following the Cranfield paradigm is costly, since it requires many instructed annotators with domain knowledge to obtain sufficient relevance feedback. Thus one can use additional data of the documents for example like citations to determine relevance. The CLEF-IP test and training set for example uses the citations of the patent examiners, which are assigned during the patent granting process, as measures of relevance [PH19]. Similarly the COLIEE test and training set use the citations of the court cases as relevant documents [RKG+20]. Another relevance signal, which requires less cost to attain the relevance labels, is implicit feedback from users. These implicit feedback signals include user behaviour like viewing of documents, time spent to view a document, clicking on a document or scrolling actions [KT03, RLS+21]. For example the test set of the TripClick collection, which we described in Section 2.5, relies on the clicks of the users. However when comparing the ranking results of used click signals to relevance assessments, Kamps et al. [KKT09] find that the different relevance measurements can lead to highly different rankings of the evaluated retrieval systems. Since human annotation is expensive, especially if large amounts of labelled data are required in order to train a neural ranking or retrieval model, another way of obtaining relevance labels is to use additional data like citations [RKG+20, PLHZ11].

The TREC Deep Learning track was the first initiative in the Information Retrieval community with the goal of comparing statistical and neural ranking and retrieval models in the large-data regime [CMYC19, CMYC20]. The organizers make the large-scale training set MS Marco available, which we described in Section 2.5, accompanied by a reliable and reusable test collection as a result of the evaluation campaign. These efforts allow training neural ranking and retrieval models for the task of ad-hoc retrieval in the web domain. The relevance labels in the MS Marco training data set come from human annotators, who are trained experts to annotate the training samples, thus the relevance labels in the training set have a high quality.

3.2.2 Crowdsourcing

Similar to annotation campaigns, crowdsourcing campaigns with non-expert annotators can be used in order to annotate training or testing datasets or to create other data resources.

Crowdsourcing offers a scalable solution for collecting annotations from a diverse group of human workers, or "crowd", to create training and test set for neural models in natural language processing and Information retrieval. Crowdsourcing platforms, such as Amazon Mechanical Turk (AMT) and Prolific, have revolutionized data annotation. Kittur et al. [KCS08] discuss how crowdsourcing harnesses collective intelligence to perform complex tasks efficiently. This collective wisdom contributes to various aspects of NLP and IR, such as document classification, entity recognition, and sentiment analysis. Poesio et al. [PCK+15] use this collective intelligence in a gamified way to create language resources through a game with players as annotators. In this game the players annotate sentences for co-reference resolution [CPK16, MBC+19, PCP+19]. For identifying which language resources need to be annotated in co-reference resolution, Madge et al. [MYC+19] propose a high-performance automatic detection model for this gamified approach of collecting crowdsourced annotations. For this gamified approach of crowdsourced annotations, Yu et al. [YPC+23] release a large-scale, annotated dataset for co-reference resolution, where they increase the size of the dataset using a resolve-and-aggregate paradigm to 'complete' markable annotations through the combination of a co-reference resolver and an aggregation method for co-reference.

While crowdsourcing offers the advantage of scale, ensuring the quality of annotations is a crucial concern. Snow et al. [SOJN08] delve into the intricacies of quality control in crowdsourced data annotation tasks. Their work highlights mechanisms for improving the accuracy and reliability of labels in NLP and IR datasets. Similarly, Aker et al. [AEAK12] investigate factors which can influence the quality of the annotations obtained from Amazon's Mechanical Turk crowdsourcing platform. They explore the effects of varied presentation methods in the annotation campaign, the location of the annotators, and payment scales on the resulting quality of the annotations. One of the intriguing findings is that their results do not align with prior studies that suggested an increase in payment attracts more noise. They also discover that the country of origin only exerts an influence in some categories and solely in general text questions, with no significant difference at the highest pay.

Efficiency in crowdsourcing experiments is also an important direction for lowering the annotation cost and increasing the efficiency of the annotators. Chamberlain et al. [CKP18] propose to increase the efficiency of crowdsourcing annotations through the implementation of a validation process, evaluated across four key parameters: quality, cost, noise, and speed. They show that an additional validation process can increase the overall quality of annotations without introducing more noise, thus a validation step can provide higher quality results than just acquiring more annotations.

Relevance judgments, critical for evaluating search engines and ranking algorithms, are often obtained through crowdsourcing [GL10, AL11, KKM11]. Grady et al. [GL10] conduct a crowdsourcing experiment for relevance annotation of search queries using amazon Mechanical Turk. In their study they measure the accuracy, time, and cost of the annotators. For accuracy

they compare the non-expert annotations to TREC NIST relevance judgements and find that the overlap between the assessors is moderate. Specifically they discuss that it is important to include a graded relevance scale rather than a binary relevance scale for annotation. Alonso et al. [AM12] compare crowdsourcing annotators from Amazon Mechanical Turk to TREC annotators. When solely comparing the relevance judgements between the two groups, they report a low agreement between the relevance judgements, however they find to improve the agreement when grouping the non-expert annotators in particular groups. They find that in cases of disagreement, the evaluations conducted by TREC assessors can be deemed at least questionable. Workers have demonstrated not only accuracy in their assessments of relevance but, in certain instances, have shown precision on par with or even surpassing that of the original expert assessors. Furthermore effective task design is crucial for obtaining accurate annotations. Kazai et al. [KKM11] examine the intricacies of designing crowdsourcing tasks for IR relevance assessment and provide guidelines for creating high-quality tasks.

In addition to explicit relevance judgments, click-through data is another valuable resource for IR evaluation [SK13, LKS12, FK01, RKJ08]. Joachims [Joa02] explore the use of crowdsourcing to generate click-through training data, emphasizing its importance in improving search engine performance. Furthermore Joachims et al. [JGP+17] investigate the trustworthiness of implicit feedback produced from click-through data within the context of web search. By scrutinizing users' decision-making processes through eye-tracking and contrasting implicit feedback with manual assessments of relevance, the authors ascertain that clicks provide valuable insights but are subject to bias. While this introduces challenges in interpreting clicks as definitive measures of relevance, their findings demonstrate that relative preferences deduced from clicks are, on average, reasonably accurate.

3.2.3 Addressing limited training data

For retrieval tasks, for which there is no large-scale training data available, there are alternate research directions on how to use neural ranking and retrieval models.

The research directions include zero-shot retrieval, where the neural ranking and retrieval models are trained on another, resource-rich, retrieval task or domain [XXS+22, TRR+21] and then solely applied to the respective retrieval task. Here the effectiveness is limited by the relatedness of the resource-rich task and the target task and the transferability of the notion of relevance from the resource-rich task to the target task.

Another direction includes few-shot learning and prompt-based in-context learning with few-shot examples [DZM+23, LLH+23, RM21]. Here a few samples for the task are labelled and are either used for fine-tuning or are directly used in the prompt of the language model to prime the generation of the language model on the few, labelled samples. There are also several studies on how to include the in-context learning samples. One technique is called Chain-of-Thought prompting [WWS+22], where the authors demonstrate a higher effectiveness of the large language model, where the model is asked to produce a chain of thought as output text. This chain of thought generation mode of the large language model gives the model the ability to decompose multi-step problems into intermediate steps and thus improves the reasoning capabilities of the

model. With an exceeding context length of large language models [Ope23, CWCT23] this direction of few-shot in-context learning is a promising new direction for adapting large language models for specific tasks.

Another approach for addressing lack of data for training neural ranking and retrieval models is generating training data using large language models [WTRG22]. Wang et al. [WTRG22] use a large generation model trained on another retrieval training dataset to generate relevant queries to a given document. These generated training samples can then be filtered to remove queries with low quality. Then the training samples can then be scored with an already trained MonoBERT model to either filter the samples or to train the final ranking model with a more fine-grained relevance distribution using knowledge distillation [HAS+20].

3.2.4 Active learning

Another direction which is not yet explored for neural ranking and retrieval models, but for early ranking models like learning-to-rank models is active learning. Active Learning seeks to optimize the efficiency of model training by reducing the cost of acquiring training labels while simultaneously maximizing the effectiveness of the trained model. The majority of Active Learning approaches have relied on two main strategies: uncertainty [LG94, Sha48] and diversity-based [SZ05]. Furthermore there are other methods including the expected gradient information in the training of the model [SCR07, SC08] or the expected performance prediction of the model [RM] for active learning. These methods have been extensively tested and validated across various learning tasks and datasets.

Active learning has been demonstrated to successfully reduce the annotation cost while improving effectiveness for various tasks in Natural Language Processing [ZSH22]. In a recent study by Schröder et al. [SNP22], these uncertainty-based strategies were revisited, particularly in the context of Transformer-based models, and they presented empirical results for text classification. They assess different query strategies on a well-established benchmark and achieve results that approach state-of-the-art performance in text classification, despite utilizing only a small portion of the training data. They also show that in contrast to prevailing belief, the prediction entropy, the supposedly strongest uncertainty-based baseline, is outperformed by several uncertainty-based strategies on this benchmark.

For measuring uncertainty for neural models, Gal et al. [GG16] have demonstrated that dropout can serve as an approximation of inference and a means to gauge model uncertainty. This deep Bayesian approach has found application in diverse natural language processing tasks [SL18, SPK+21].

Another active learning method explored for natural language processing tasks is using the gradient information first introduced by [SCR07]. Here the training sample is chosen with the highest impact on the weights of the model that is trained. Settles et al. [SC08] apply this strategy for sequence labeling and study it in the context of sequence labeling benchmarks. Zhang et al. [ZLW17] investigate a variant of expected gradient length designed for neural networks, focusing solely on word embedding gradients and demonstrating its efficacy in text classification.

For diversity-based selection, multiple studies in the field of natural language processing investigate active learning based on diversity [XYT+03, Zhd19, YKZ+22, MZK+22]. Maekawa et al. [MZK+22] investigate the effectiveness of current active learning approaches in the context of interactive labeling within a low-resource setting. Their experiments show that existing methods frequently yield sub-optimal results in the specific scenario, with increased response times of annotators and limited adaptability. To address these issues, they propose a novel active learning technique, which combines hybrid sampling strategies to reduce labeling costs and acquisition delays, while offering the flexibility to adapt to dataset variability through user-guided interactions.

Active Learning has been applied to several tasks and scenarios in the field of Information Retrieval [CGJ96, LC94, SOS92, YWGH09, ZWYT08, LBC+15, DC09, SGV14, DC08]. Cai et al. [CGZW11] employ a transfer learning approach, where they adapt a learning-to-rank (LTR) model trained on one domain for use in another domain. They employ the Query-by-Committee (QBC) algorithm [SOS92] for active query selection during domain adaptation. QBC is used to intelligently select queries from both the target and source domains, as well as to mix their respective training sets. This strategy results in more effective domain adaptation for LTR models compared to random query selection, especially when training data is limited. Xu et al. [XAZ07] explore active learning strategies that emphasize diversity in updating query relevance scoring. They propose a combination of diversity and density-based selection for enhancing LTR models. However, the use of Active Learning in the context of fine-tuning neural rankers has not been explored until now.

A variation of the Active Learning setting that has shown success in certain domain-specific tasks is that of continuous active learning [GC11, YMLF22, SC22], where documents are iteratively retrieved by actively learning for one specific query, typically aiming for total recall [GCR16]. For the task of technology assisted review (TAR), Yang et al. [YLF21] propose a TAR cost framework, however this framework focuses on cost modeling for reviewing one specific query.

The effect of the size of training data available for a task has been observed in the context of training neural ranking and retrieval models. Previous studies find that decreasing the training data size significantly decreases the ranking or retrieval effectiveness of a neural ranking or retrieval model in the web domain [KOM+20, GM22, FAPH22, ZXM+22, CMY+21a] as well for domain-specific retrieval tasks [HYX+22, GC21, WTRG22, MBB21]. Nogueira et al. [NJPL20] observed variations in the effectiveness of fine-tuning a MonoBERT ranker on subsets of the training data of different sizes. Iurii et al. [MBB21] investigated transfer learning for MonoBERT rankers by first fine-tuning them on MS MARCO and then transferring them to question-answering tasks in both a zero-shot and full training setting. They examined the impact of training on subsets of the training data and found that the effectiveness increased as the number of training queries increased. They also noted that the source and target domains had large training datasets. Zhang et al. [ZYL20] explore domain transfer of BERT cross-encoders in a situation with limited data availability. Specifically, they investigate the transfer of MonoBERT from web search (trained on MS Marco) to small, domain-specific retrieval tasks. Surprisingly, they found that using small in-domain training data sometimes reduced search effectiveness compared to the zero-shot application of MonoBERT.

This research demonstrates that it is critical for a highly effective neural ranking or retrieval model, to be trained on a large-scale, high quality training dataset.

3.2.5 Summary

Overall this related work shows that lack of reliable evaluation data and lack of training data or cost efficient methods for acquiring training data are open, important research fields. There has been extensive work on creating resources for testing and comparing ranking and retrieval systems in evaluation campaigns. We have described different evaluation campaigns which create resources for a variety of retrieval tasks. Another approach for addressing the lack of data for evaluation and training ranking models is to use crowdsourcing. We have introduced related work on crowdsourcing for attaining datasets for testing and training ranking and retrieval models. Furthermore we have described approaches to address limited training data, like zero-shot application of ranking models, few-shot and in-context learning as well as generating labels using large language models. In more detail we describe related work in active learning in the Natural Language Processing and Information Retrieval community. Overall pursuing these research directions of addressing lack of data for training and testing ranking and retrieval models can be advantageous for numerous stakeholders in the Information Retrieval community in the long term, and it may initiate a diverse set of subsequent studies.

CHAPTER 4

Neural Ranking and Retrieval for Document-to-Document Retrieval

As we show with **Search Example ❶** and **Search Example ❷** there are document-to-document retrieval tasks in specific domains, that require an effective and exhaustive search process for finding all relevant evidence. We investigate in this section, how to adapt neural ranking and retrieval models for document-to-document retrieval tasks and address the research question:

RQ1 How can neural ranking and retrieval models be adapted for document-to-document retrieval tasks?

We study this research question in the context of the prior case retrieval task in the legal domain and prior art search in the patent domain. In the first subchapter we investigate how neural ranking architectures can be adapted for document-to-document retrieval tasks and then focus on the adaptation of neural retrieval models in the second subchapter.

4.1 Paragraph-Level Interaction Re-Ranking for Document-to-Document Retrieval

This chapter is based on the publication [AHH21].

In the last years, pre-trained language models – such as BERT – revolutionized web and news search [Nay19, NC19]. Naturally, the community aims to adapt these advancements to cross-domain transfer of retrieval models for domain specific search. In the context of legal case retrieval, Shao et al. propose the BERT-PLI framework by modeling the **P**aragraph-**L**evel **I**nteractions with the language model BERT [SML+20]. In order to investigate how we can adapt neural ranking models for document-to-document retrieval tasks, we reproduce the original experiments, we clarify pre-processing steps and add missing scripts for framework steps, however we are not able to reproduce the evaluation results. Contrary to the original paper, we demonstrate that the domain

specific paragraph-level modelling does not appear to help the performance of the BERT-PLI model compared to paragraph-level modelling with the original BERT. In addition to our legal case retrieval reproducibility study, we investigate BERT-PLI for prior art search in the patent domain. We find that the BERT-PLI model does not yet achieve performance improvements for patent document retrieval compared to the BM25 baseline. Furthermore, we evaluate the BERT-PLI model for cross-domain retrieval between the legal and patent domain on individual components, both on a paragraph and document-level. We find that the transfer of the BERT-PLI model on the paragraph-level leads to comparable results between both domains as well as first promising results for the cross-domain transfer on the document-level. For reproducibility and transparency as well as to benefit the community we make our source code and the trained models publicly available.

4.1.1 Introduction

Bringing the substantial effectiveness gains from contextualized language retrieval models from web and news search to other domains is paramount to the equitable use of machine learning models in Information Retrieval (IR). The promise of these pre-trained models is a cross-domain transfer with limited in-domain training data. Thus we investigate in this work the document retrieval on two specific language domains, the legal and the patent domain, and study the transferability of the retrieval models between both domains.

In case law systems the precedent cases are a key source for lawyers, therefore it is essential for the lawyers' work to retrieve prior cases which support the query case. Similarly in the patent domain, patent examiners review patent applications and search for prior art, in order to determine what contribution the invention makes over the prior art. The recent advances in language modelling have shown that contextualized language models enhance the performance of information retrieval models in the web and news domain compared to traditional ad-hoc retrieval models [HH19, HZH20a]. However for legal and patent retrieval we have a different task setting as the documents contain longer text with a mean of 11,100 words per document [RGK$^+$22]. In document retrieval every passage may be relevant, therefore in a high-recall setting such as ours it is crucial for the retrieval model to take the whole document into account. This is a challenge for contextualized language retrieval models, which are only capable of computing short passages with a length up to 512 tokens [GDC20, YXL$^+$19, NZG$^+$20].

Recently, Shao et al. [SML$^+$20] aimed to bring the gains of language modelling to legal document retrieval and tackle the challenge of long documents by proposing BERT-PLI, a multi-stage framework which models **P**aragraph-**L**evel **I**nteractions of queries and candidates with multiple paragraphs using BERT [DCLT19]. The document-level relevance of each query and candidate pair is predicted based on paragraph-level interaction of the query and candidate paragraphs which are aggregated with a recurrent neural network (LSTM or GRU). The BERT-PLI model is trained in two stages: first, BERT is trained on a paragraph entailment task, and second the recurrent aggregation component is trained on a binary classification task.

In order to answer the research question:

RQ1.1 How can neural ranking models be adapted for document-to-document retrieval tasks?

, we reproduce the results of BERT-PLI for the legal retrieval task and extend the study by training and evaluating BERT-PLI for prior art search in the patent domain.

For the reproduction on the legal case retrieval task, we found shortcomings in the description of the data pre-processing and evaluation methods, after a discussion with the authors of the original paper we could clarify how the evaluation results are achieved. As the published code is missing crucial parts, we re-implement the preprocessing, the first stage BERT fine-tuning as well as the retrieval with BM25 in the second stage and the overall evaluation. Furthermore we analyze the ablation study of the original paper and answer the following research question:

RQ1.1.1 Does fine-tuning BERT on domain specific paragraphs improve the retrieval performance for document retrieval?

The original paper finds a $7 - 9\%$ performance improvement of the BERT-PLI model for legal retrieval, when fine-tuning BERT on the legal paragraphs. Contrary to the original paper, we find that the paragraph-level modelling with BERT, fine-tuned on the domain specific paragraph-level modelling, does not appear to help the BERT-PLI model's performance on legal document retrieval. In line with that, we also demonstrate that the patent specific paragraph-level modelling harms the performance of the BERT-PLI model also for the patent retrieval task and remains a promising opportunity.

In order to analyze the proposed BERT-PLI model for another document retrieval task with long documents, we investigate following research question:

RQ1.1.2 To what extent is a BERT-PLI model, which is trained on patent retrieval, beneficial for document retrieval in the patent domain?

We find that the patent domain BERT-PLI model is outperformed by the BM25 baseline for the patent retrieval task. This shows that the document retrieval with BERT is not yet beneficial for the patent retrieval and stays a promising opportunity.

As the legal and patent documents come from similar language domains, it becomes an interesting question to what extent we can transfer the domain specific retrieval models from one to the other domain. Especially because of the restricted accessibility of domain specific, labelled retrieval data there is the need for studying cross-domain transfer of document retrieval models.

RQ1.1.3 To what extent is cross-domain transfer on paragraph- and document-level of the domain specific BERT-PLI model between legal and patent domain possible?

We show that the transfer of the domain specific paragraph-level interaction modelling is possible between the legal and patent domain with similar performance of the retrieval model. Furthermore we find on the document-level transfer that the zero-shot application of a patent domain specific BERT-PLI model for the legal retrieval task achieves a lower performance than the BM25 baseline. Showing first promising results, the cross-domain transfer of retrieval models stays an open and exciting research direction. Our main contributions are:

- We reproduce the experiments of Shao et al. [SML+20] and investigate shortcomings in the data pre-processing and model methods. Contrary to the original paper we find that

domain specific paragraph-level modelling does not appear to help the performance of the BERT-PLI model for legal document retrieval

- We train a domain specific BERT-PLI model for the patent domain and demonstrate that it does not yet outperform the BM25 baseline
- We analyze the cross-domain transfer of the BERT-PLI model between the legal and patent domain with first promising results
- In order to make our results available for reproduction and to benefit the community, we publish the source code and trained models in the Github repository linked in Section 1.3

4.1.2 Methods

Task description

Document retrieval in the legal and patent domain are specialized IR tasks with the particularity that query and candidates are long documents which use domain specific language.
In legal document retrieval, the relevant documents are defined as the previous cases which should be noticed for solving the query case [RKG+20], in other words which support or contradict the query document [SML+20]. The legal documents consist of long text containing the factual description of a case.
Relevance in the patent domain is defined for the prior art search task [PLHZ11], i.e. it is the task to find documents in the corpus that are related to the new invention or describe the same invention. The patent documents consist of a title, an abstract, claims and a description as well as metadata like the authors or topical classifications. As we investigate retrieval and classification based on the textual information, we will only consider the textual data of the patent documents.

BERT-PLI architecture overview

As the BERT model advanced the state-of-the-art in natural language processing and information retrieval, but has the restriction that it can only model the relation between short paragraphs, Shao et al. [SML+20] propose a multi-stage framework model using BERT for the retrieval of long documents which is illustrated in Figure 4.1. The training is separated into two stages. In stage 1, a MonoBERT re-ranking model, as introduced in Section 2.6.2 is fine-tuned on a relevance prediction task on a paragraph-level. MonoBERT takes the concatenated query and document paragraph as input and is then fine-tuned on predicting the relevance of the candidate paragraph to the query paragraph given the output vector of the special [CLS] token of BERT. Therefore this output vector is trained to be a relevance representation on a paragraph-level of the two concatenated input paragraphs.

This fine-tuned MonoBERT ranker is used in stage 2, where the full document retrieval with paragraph-level interaction modelling takes place. For a query document q the top K candidates are retrieved from a corpus using BM25 [RZ09], and the query document as well as the top K candidates are split into paragraphs. Then for each candidate $i \in 1, .., K$ the first N paragraphs of the query document and the first M paragraphs of the candidate are concatenated and their

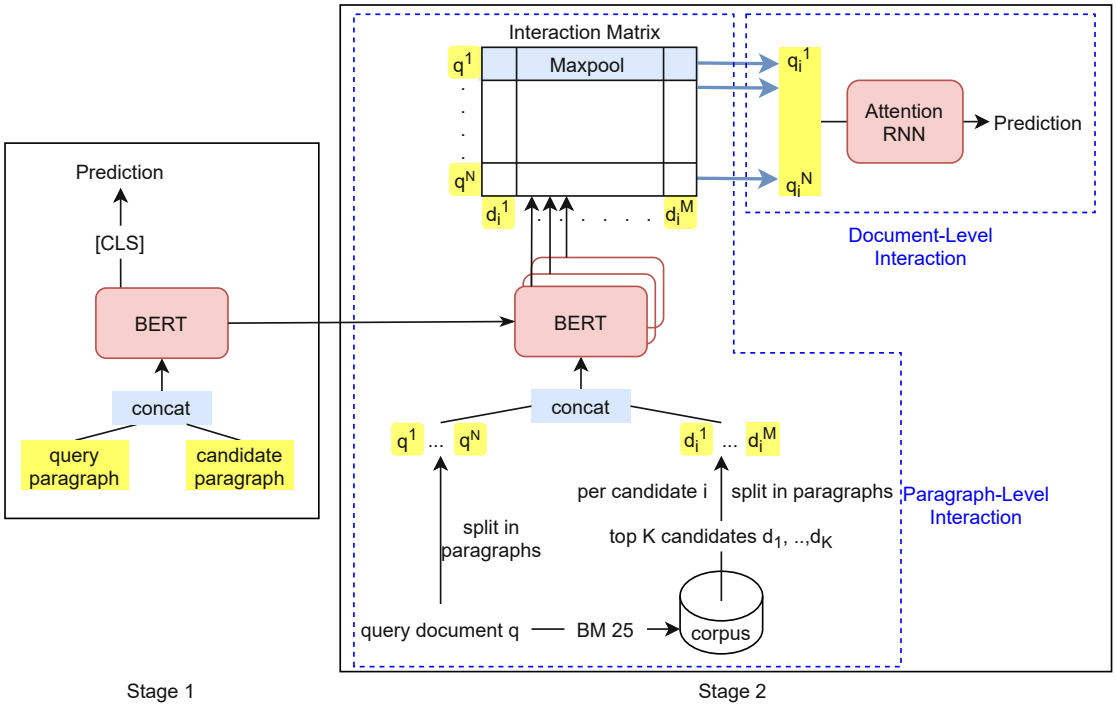


Figure 4.1: BERT-PLI Multistage architecture

relevance representation is calculated with the BERT model from stage 1. This yields an interaction matrix between the query and candidate paragraphs. An additional Maxpooling layer captures the strongest matching signals per query paragraph and yields a document-level relevance representation of the query and the candidate. This document-level relevance representation is used to train an RNN model with a succeeding attention and fully-connected forward layer which we will refer to as Attention RNN. This Attention RNN yields the binary prediction of the relevance for the query and candidate document.

Cross-domain evaluation approach

In the first stage of the BERT-PLI framework the BERT model learns to model the paragraph-level interaction. For the two different domains we fine-tune the BERT model on a paragraph-level relevance prediction task, which yields the paragraph-level interaction **LawBERT** model for the legal and the **PatentBERT** model for the patent domain. In order to analyze the influence of the domain specific paragraph-level modelling, we compare the document retrieval models trained with the paragraph-level modelling of LawBERT or PatentBERT to document retrieval models trained on the paragraph-level modelling of the original BERT model. The paragraph-level modelling with the original BERT model is denoted with $\mathbf{BERT_{ORG}}$ as in Figure 4.2. Based on these paragraph-level interaction representations we train an AttentionRNN on the legal as well as on the patent document-level retrieval task, which we denote with **LawRNN** or **PatentRNN** respectively. In order to isolate the impact of the different modelling of the paragraph-level

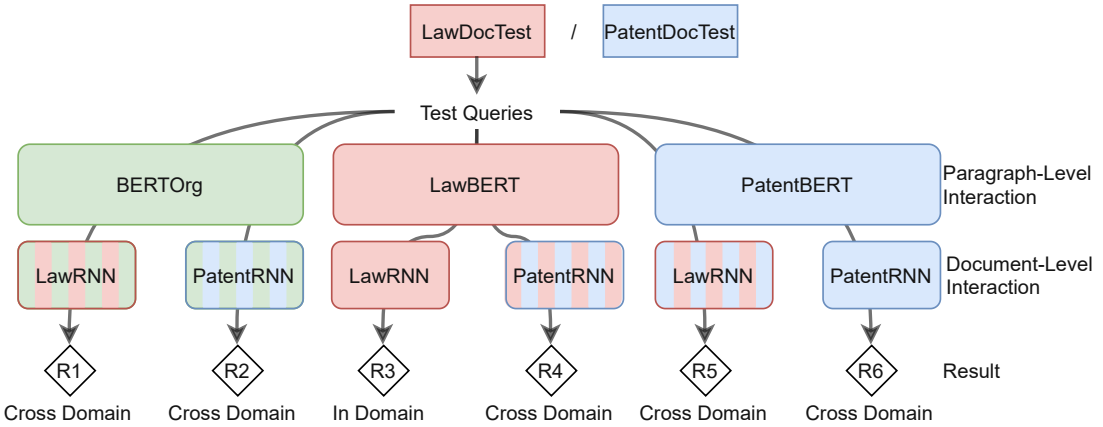


Figure 4.2: Cross-domain evaluation approach

interactions from LawBERT and PatentBERT, we additionally train an AttentionRNN on the patent document retrieval task given their LawBERT relevance representations and vice versa. We evaluate the resulting models on the legal or the patent test document retrieval set, namely **LawDocTest** or **PatentDocTest**. This process is visualized in Figure 4.2 and yields six evaluation results R1-6 for each test set. For example for LawDocTest, R3 is the in-domain evaluation result, whereas the other results denote cross domain evaluations. For LawDocTest the results R1, R3 and R5 are all from LawRNN document retrieval, but the LawRNNs differ in the paragraph-level relevance representation they are trained with. Therefore comparison of the results R1, R3 and R5 on LawDocTest shows the transferability of the paragraph-level modelling between the legal and patent domains and the difference of domain-specific paragraph-level modelling to the non-domain specific modelling. Furthermore to analyze the cross-domain transfer on the document-level, we compare the evaluation results of LawDocTest and PatentDocTest of R1 and R2, R3 and R4 as well as R5 and R6. This comparison shows the cross-domain transferability on the document-level as the LawRNN and PatentRNN share the same paragraph-level relevance representations, which they are trained on.

4.1.3 Experiments

Datasets

Legal retrieval dataset Like Shao et al. [SML+20], we use the legal retrieval collections from the COLIEE evaluation campaign 2019 [RKG+20] introduced in section 2.5, which consist of a paragraph-level and a document-level retrieval task. Both retrieval collections are based on cases from the Canadian case law system and are written in English. The paragraph-level task (COLIEE 2019 Task 2) involves the identification of a paragraph which entails the given query paragraph [RKG+20]. For this task the COLIEE evaluation campaign provides training and test queries with relevance judgements which we will refer to as **LawParaTrain** and **LawParaTest**. In the document-retrieval task (COLIEE 2019 Task 1) it is asked to find supporting cases from a provided set of candidate documents, which support the decision of the query document. As

Table 4.1: Statistics of the training and test set for the paragraph the document-level retrieval task

	LawPara		LawDoc		PatentPara		PatentDoc	
	Train	Test	Train	Test	Train	Test	Train	Test
# of queries	181	41	285	61	44	42	351	100
avg # of candidates	32.12	32.19	200	200	3.5M	3.5M	3.5M	3.5M
avg # relevant candidates	1.12	1.02	5.21	5.41	43.52	76.3	3.27	2.85

in the original paper we take 20% of the queries of the training set as validation set, denoted with **LawDocVal**. We will refer to the training and test datasets for the document retrieval as **LawDocTrain** and **LawDocTest**.

Patent retrieval dataset For the patent retrieval queries and relevance judgments we use the datasets from the CLEF-IP evaluation campaign [PLH13] introduced in section 2.5. We choose the CLEF-IP collection since it provides a patent corpus and training and test collections for patent retrieval tasks on the paragraph- and document-level. The tasks contain English, French and German queries, we only consider the English queries and candidates. For the paragraph-level training and test collection we choose the provided queries and relevance judgements from the passage retrieval task starting from claims of the CLEF-IP 2013 [PLH13] where the participants are asked to find passages from patent documents which are relevant to a given set of claims. We refer to these datasets as **PatentParaTrain** and **PatentParaTest**. As the document-level training and test collection we choose the queries and relevance judgements from the prior art candidate search from the CLEF-IP evaluation campaign 2011 [PLHZ11] and refer to them as **PatentDocTrain** and **PatentDocTest**. As in the original paper, we take 20% of the training set as validation set, denoted with **PatentDocVal**. Both patent retrieval tasks retrieve paragraphs and documents from the patent corpus which consists of 3.5 million patent documents filed at the European Patent Office (EPO) or at the World Intellectual Property Office (WIPO).
The dataset statistics can be found in Table 4.1.

Experiment setting

We conceptualize the training of the BERT-PLI model in stage 1 (paragraph-level interaction fine-tuning) and stage 2 (document retrieval) in the pseudo-code 1. In the pseudo-code we use the example of training BERT-PLI for legal case retrieval and thus on the legal datasets, described in the previous section, however this process is the same for training BERT-PLI on the prior art search task with the difference of using the datasets for the respective task.

Stage 1: MonoBERT fine-tuning

In the first stage we fine-tune the MonoBERT model[1] on the paragraph-level relevance ranking for either the legal domain or the patent domain to attain LawBERT and PatentBERT. As there was no code open-sourced for fine-tuning BERT, we use the HuggingFace transformers library[2] and

[1] Checkpoint from https://github.com/google-research/bert.
[2] https://github.com/huggingface/transformers

Algorithm 1 Training process of BERT-PLI with Stage 1 (paragraph-level interaction) and Stage 2 (document retrieval) on the example of training BERT-PLI for legal case retrieval

Input: **LawParaTrain** paragraph-level training set, **LawDocTrain** document-level training set, **M** MonoBERT (not trained), $BM25$ BM25 model, $ARNN$ AttentionRNN Layer of BERT-PLI, MP Max-Pooling layer of BERT-PLI, e_{para} number of epochs for paragraph-level training, e_{doc} number of epochs for document-level training, N number of query paragraphs in BERT-PLI, M number of document paragraphs in BERT-PLI

Output: M MonoBERT trained on **LawParaTrain**, $ARNN$ AttentionRNN trained on **LawDocTrain**

Stage 1 Paragraph-level training
for n *in* e_{para} **do**
  Concatenate query and candidate paragraph from **LawParaTrain** to training sample t
  Train M with t
  end
Stage 2 Document-level training
Index documents of **LawDocTrain** in inverted index I
Retrieve top 50 documents D for queries in **LawDocTrain** from I with $BM25$
Split queries in **LawDocTrain** and candidate documents D into paragraphs
Take first N query paragraphs q^1, .., q^N and first M document paragraphs d^1, .., d^M
Concatenate all q^i, d^j paragraphs for $i \in N$, $j \in M$ and score with M
MP layer to get query-document representations q_i^1, .., q_i^N
for n *in* e_{doc} **do**
  Train $ARNN$ with q_i^1, .., q_i^N
  end

add the MonoBERT fine-tuning script to the published code. Different to the MonoBERT ranking training with RankNetloss, we use the Binary Cross entropy loss for fine-tuning MonoBERT on the task of classifying relevant documents as relevant and irrelevant document as irrelevant. We use this loss as Shao et al. [SML+20] also describe the fine-tuning of MonoBERT with the Binary Cross entropy loss.

For LawBERT we use the LawParaTrain as training and LawParaTest as test queries and relevance judgements. In order to use the queries and relevance judgements for a binary classification task, we consider the paragraph pairs of the query and one relevant candidate as positive samples. It was not stated clearly in the original paper how the paragraph pairs of negative samples are constructed, therefore we investigate this data pre-processing decision. We find that taking all paragraph pairs constructed of the query and a non-relevant paragraph from the paragraph candidates as negatives, yields comparable results for fine-tuning the BERT model on the legal domain as in the original paper. This negative sampling approach results in 3% positive and 97% negative samples in the training set. The queries and paragraph candidates have less than 100 words on average and are truncated symmetrically if they exceed the maximum input length of 512 tokens of BERT. For the training batch size we do a grid search and find that the F1-score of LawParaTest is the highest with a batch size of 2 (65.1% F1-Score) instead of 1 (63.4% F1-Score)

after fine-tuning BERT for 3 epochs on LawParaTrain, contrary to the original paper: they report the highest F1-score of 65.2% without reporting the batch size. As stated in the original code, we assumed they used the batch size of 1, due to our comparison we use a batch size of 2 instead of 1. After a remark of the original authors it turns out the original implementation was done with a batch size of 16. For the learning rate we also do a grid search and find that the learning rate of $1e-5$ is optimal as in the original paper. As in the original paper, we fine-tune for 3 epochs and we do the final fine-tuning of the LawBERT model on the merged training and test set. This is permissible as we train and evaluate the BERT-PLI model on LawDocTrain and LawDocTest, the LawParaTrain and LawParaTest sets are only used for fine-tuning LawBERT.
For the PatentBERT fine-tuning we use the PatentParaTrain as training and PatentParaTest as test set. We construct the negative paragraph pairs by sampling randomly paragraphs (which are not the relevant paragraph) from the documents which contain a relevant paragraph to a query paragraph. Here we sample randomly 5 times the number of positive paragraphs as negatives, as otherwise the share of positive pairs is below 1% and in order to have a similar ratio as for the legal domain. We do a grid search for the training batch size and learning rate and find that a batch size of 2 with a learning rate of $2e-5$ yields the highest F1-score of 19.0%. We fine-tune PatentBERT solely on PatentParaTrain as it is practice in machine learning to hold out the test set.

Stage 2: Document retrieval

In stage 2 the first step is to retrieve relevant documents from the given set of candidates (in the legal domain) or from the whole corpus (in the patent domain). As it was not clearly stated in the original paper nor was there code published, how to employ the BM25 algorithm [RZ09] for this first step, we re-implement this step and use the BM25 algorithm with $k_1 = 0.9$ and $b = 0.4$ implemented in the Pyserini toolkit[3]. Furthermore we do a grid search for the input length to the BM25 algorithm and find that the top $K = 50$ retrieval with input length of 250 leads to similar recall scores as the original paper for the LawDocTrain set (93.22%) and the LawDocTest (92.23%). Here we only consider recall scores as in the original paper, as the focus of the first step BM25 retrieval is to retrieve all relevant cases for re-ranking for the training and test set.
For patent document retrieval, the task is to retrieve relevant documents from the patent corpus with 3.5 million documents. As in the patent document retrieval task only 3.27 relevant patent documents per query document are contained and as the recall does not significantly increase when taking $K = 50$ candidates, we choose the top $K = 20$ from the BM25 retrieval, in order to have a similar ratio of positive and negative pairs as in the legal document retrieval for training the AttentionRNN. Here we find that the BM25 algorithm with the document input length of 250 reaches the highest recall score of 9.42% on PatentDocTrain compared to other document input lengths. Due to the low recall score of the retrieved documents on PatentDocTrain we add the relevant documents from the relevance judgements to PatentDocTrain and sample randomly non-relevant documents from the BM25 candidates for the training dataset, so that we have in total 20 candidates. For PatentDocTest we retrieve the top 50 candidates as in the original implementation where we reach a recall of 10.66%, but we do not add the relevant candidates after the BM25 retrieval step. In order to reproduce the experiments for modelling the

[3] https://github.com/castorini/pyserini

paragraph-level interaction and training the Attention RNN, we use the open-sourced repository[4] of the original paper. As in the original paper we set the number of paragraphs of the query $N = 54$ and the number of paragraphs of the candidate $M = 40$ for legal and patent retrieval. The query and candidate documents are split up in paragraphs of 256 tokens. We model the paragraph-level interactions of LawDocTrain, LawDocTest, PatentDocTrain and PatentDocTest using LawBERT or PatentBERT or BERT_{ORG}. With these paragraph-level representations of each query and its candidate document we train an AttentionRNN network with either an LSTM [HS97] or a GRU network [CvMG+14] as RNN on classifying the relevance between the query and candidate document. The AttentionRNN trained on the LawDocTrain is denoted with LawRNN, on PatentDocTrain it is denoted with PatentRNN. For training the AttentionRNN we use the same hyperparameter as in the original implementation, except for the PatentBERT LawRNN configuration, where we find that the learning rate of $1e-4$ is better suited, when evaluated on the LawDocVal set.

4.1.4 Evaluation and Analysis

In-domain evaluation for legal document retrieval (RQ1.1.1)

Shao et al. [SML+20] evaluate their models using the binary classification metrics precision, recall and F1-Score on the whole test set. Furthermore they compare their model performance to the two best runs from the COLIEE 2019 denoted by the team names JNLP [TNS19] and ILPS [RK19]. As it was not clearly stated in the original paper, we assume that Shao et al. [SML+20] evaluate the BERT-PLI models on the whole LawDocTest set with all 200 given candidates per query. With the first retrieval step, the top 50 query candidate pairs are retrieved for binary classification, therefore we assume the lower 150 candidates classified as irrelevant. As in [SML+20], we use a cutoff value of 5 for the evaluation of ranking algorithms like BM25, this means the top 5 retrieved documents are classified as relevant, whereas the remaining 195 are considered irrelevant.
As Shao et al. [SML+20] evaluate in their published code the top 50 candidates, we investigate the overall evaluation of our reproduced BERT-PLI models for all 200 candidates with the precision, recall and F1-score using the SciKitlearn classification report [5]. The results can be found in Table 4.2, we test the statistical significance compared to the BM25 baseline with the Student's paired, independent t-test [SAC07, ULH19]. Comparing the evaluation results stated in the original paper and our evaluation results, we find that our reproduced BERT-PLI LawBERT LSTM and GRU model reach similar values. On the effect of domain specific paragraph-level modelling on the legal case retrieval task (RQ1.1.1), the original paper reports a $7-9\%$ performance improvement for legal retrieval with the BERT-PLI model, when BERT is fine-tuned on the legal paragraph-level modelling compared to the original BERT. Contrary to that, we find that the domain specific paragraph-level modelling does not appear to help the performance of the legal case retrieval. Our reproduced BERT_{ORG} LawRNN GRU model outperforms all other BERT-PLI models except on the recall, however this shows that contrary to the findings in the original paper,

[4] https://github.com/ThuYShao/BERT-PLI-IJCAI2020
[5] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

Table 4.2: Precision, Recall and F1-Score comparison of Shao et al. [SML$^+$20] and our reproduction, BM25 cutoff value of 5 as in [SML$^+$20], JNLP [TNS19] and ILPS [RK19] denote the best two runs of the COLIEE 2019, † indicates statistically significant difference to BM25, $\alpha = 0.05$

Team/Model	Precision	Recall	F1-Score
JNLP [TNS19]	0.6000	0.5545	0.5764
ILPS [RK19]	0.68	0.43	0.53
BERT_{ORG} LawRNN LSTM [SML$^+$20]	0.5278	0.4606	0.4919
BERT_{ORG} LawRNN GRU [SML$^+$20]	0.4958	0.5364	0.5153
LawBERT LawRNN LSTM [SML$^+$20]	0.5931	0.5697	0.5812
LawBERT LawRNN GRU [SML$^+$20]	**0.6026**	**0.5697**	**0.5857**
Reproduction			
BM25 (cutoff at 5)	0.5114	0.5360	0.5234
Repr BERT_{ORG} LawRNN LSTM	0.7053†	0.5017†	0.5863†
Repr BERT_{ORG} LawRNN GRU	**0.8972**†	0.4501†	**0.5995**†
Repr LawBERT LawRNN LSTM	0.8620†	0.4295†	0.5733†
Repr LawBERT LawRNN GRU	0.3826†	**0.6838**†	0.4907†

the domain specific paragraph-level modelling does not always improve the performance of the BERT-PLI model.

In-domain evaluation for patent document retrieval (RQ1.1.2)

In order to investigate the applicability of the BERT-PLI model for information retrieval in the patent domain, we evaluate the PatentBERT PatentRNN models trained on PatentDocTrain. The results can be found in Table 4.3, now we analyze the in-domain evaluation for the PatentBERT PatentRNN models on PatentDocTest. This shows that the in-domain, patent BERT-PLI model is not beneficial for patent document retrieval, as it is outperformed by the BM25 baseline on all metrics. We reason this could be due to the number of considered query and candidate paragraphs (N and M), which is fit to the legal retrieval but not to the patent retrieval and could be unsuitable for patent retrieval as PatentDocTrain and PatentDocTest contain on average more paragraphs than LawDocTrain and LawDocTest. This demonstrates that the document retrieval with contextualized language models for the patent domain is not yet beneficial and needs to be taken under further investigation. In line with the findings regarding RQ1.1.1 for the legal document retrieval, we find that the paragraph-level modelling with the PatentBERT model impairs the performance of the document retrieval compared to the paragraph-level modelling with BERT_{ORG}. This shows that the domain specific paragraph-level modelling is not always beneficial for BERT-PLI for the legal and patent document retrieval.

Cross-domain evaluation (RQ1.1.3)

In order to analyze the cross-domain retrieval between the legal and patent domain, we evaluate each model on LawDocTest and PatentDocTest set as illustrated in Figure 4.2 and compare for each test set the performance of the different models in order to gain insights about the

Table 4.3: In-domain and cross-domain evaluation on the legal and patent document retrieval test set, in-domain evaluation for LawBERT LawRNN models on LawDocTest and PatentBERT PatentRNN on PatentDocTest, R1-6 denote the result numbers from Figure 4.2, † indicates statistically significant difference to BM25, $\alpha = 0.05$

		LawDocTest			PatentDocTest		
Model		Prec	Rec	F1	Prec	Rec	F1
In-domain							
BM25 (cutoff at 5)		0.5114	0.5360	0.5234	**0.0500**	**0.3968**	**0.0888**
LawBERT LawRNN (R3)	LSTM	**0.8620**†	0.4295†	**0.5733**†	0.0207†	0.4761†	0.0398†
	GRU	0.3826†	**0.6838**†	0.4907†	0.0181†	0.4444†	0.0349†
PatentBERT PatentRNN (R6)	LSTM	0.7500†	0.2268†	0.3482†	0.0365†	0.1904†	0.0613†
	GRU	0.1153†	0.0412†	0.0607†	0.0416†	0.1904†	0.0683†
Cross-domain							
LawBERT PatentRNN (R4)	LSTM	0.1103†	0.5292†	0.1826†	0.0277†	0.1587†	0.0472†
	GRU	0.0961†	0.2749†	0.1424†	0.0246†	0.1904†	0.0436†
PatentBERT LawRNN (R5)	LSTM	0.8000†	0.4673†	0.5900†	0.0188†	0.3650†	0.0357†
	GRU	0.5460†	0.5704†	0.5579†	0.0233†	0.5555†	0.0448†
BERTOrg PatentRNN (R2)	LSTM	0.0000†	0.0000†	0.0000†	0.0602†	0.0793†	0.0684†
	GRU	0.0000†	0.0000†	0.0000†	**0.0769**†	0.0952†	0.0851†
BERTOrg LawRNN (R1)	LSTM	0.7053†	0.5017†	0.5863†	0.0160†	**0.8095**†	0.0314†
	GRU	**0.8972**†	0.4501†	**0.5995**†	0.0199†	0.4285†	0.0381†

transferability of the models between the legal and patent retrieval task and on the paragraph as well as on the document-level.

Analyzing the cross-domain transfer on the paragraph-level for LawDocTest, we see in Table 4.3 that the performance is similar for the LawRNNs when modelling the paragraph-level interaction with PatentBERT instead of LawBERT. An interesting result is the performance of the PatentBERT PatentRNN LSTM model, which was not trained on modelling legal paragraph-interactions nor legal document retrieval, but performs well on LawDocTest, however it does not outperform the domain independent BM25 baseline. On the document-level we see that the PatentRNN models have on average a 40% lower F1-Score than the LawRNN models with the same paragraph-level modelling, although we see a positive effect of modelling the paragraph-level interactions with BERT_{ORG} instead of LawBERT or PatentBERT.

For the cross-domain evaluation on PatentDocTest, we find that each BERT-PLI model is outperformed by the BM25 baseline, except for the precision of the BERT_{ORG} PatentRNN models and the recall of the BERT_{ORG} LawRNN models. On the document-level transfer we see a consistent performance improvement of the PatentRNN models compared to the LawRNN models independent of the paragraph-level modelling, which leads to the conclusion that the domain specific training for patent document retrieval is beneficial here. On a paragraph-level transfer we can see a similar performance of the LawRNN models, independent of the paragraph-level modelling. For the PatentRNN models we find that the paragraph-level modelling with BERT_{ORG}

outperforms the modelling with PatentBERT and LawBERT.

4.1.5 Conclusion

We reproduce the BERT-PLI model of Shao et al. [SML+20] for the legal document retrieval task of the COLIEE evaluation campaign 2019 [RKG+20]. We have addressed shortcomings of the description of the data pre-processing and the second stage retrieval, which we investigated and for which we complemented the published code. Contrary to the original paper, we find that modelling the paragraph-level interactions with a BERT model fine-tuned on the domain does not appear to help the performance of the BERT-PLI model for document retrieval compared to modelling the paragraph-level interactions with the original BERT model. Furthermore we have analyzed the applicability of the BERT-PLI model for document retrieval in the patent domain, but we find that the BERT-PLI model does not yet improve the patent document retrieval compared to the BM25 baseline. We reason that the optimal number of query and candidate paragraphs to be considered for the interaction modelling could be a decisive hyperparameter to take into account. However bringing the gains from contextualized language model to patent document retrieval stays an open problem. We have investigated to what extend the BERT-PLI model is transferable between the legal and patent domain on the paragraph and document-level by evaluating the cross-domain retrieval of the BERT-PLI model. We show that the cross-domain transfer on the paragraph-level yields comparable performance between the legal and the patent domain. Furthermore the comparison on the document-level transfer shows first promising results when applying the BERT-PLI model trained on the patent domain to the legal domain. How to bring the benefits of contextualized language models to domain-specific search and how to transfer retrieval models across different domains remain open and exciting questions.

Limitations and future work

One limitation in this study is the focus on English. The legal case retrieval task exists in many different national legislations [MSW+21], thus the documents are also written in different languages rather than English. Furthermore for the task of prior art search in patent retrieval, the focus on English language is another limitation. The same patent can exist in many different languages and patents can also be multilingual, thus it is important for future work to investigate these findings for other languages or multilingual documents.

Another limitation is the focus on the language model BERT [DCLT19] for contextualizing and embedding the language representation for ranking. There are many different language models [BMR+20, RSR+20, LOG+19], which could be used as bases for the paragraph-level interaction model. Future work is needed to investigate how the choice of large language model influences the effectiveness of the BERT-PLI model.

4.2 Passage-Aggregation Retrieval Model for Document-to-Document Retrieval

After investigating the adaptation of neural re-ranking models for document-to-document retrieval tasks, we investigate in this chapter:

RQ1.2 How can neural retrieval models be adapted for document-to-document retrieval tasks?

This chapter is based on the publication [AHS+22].

Neural first stage retrieval models, like dense passage retrieval models (DPR) [KOM+20], show great effectiveness gains in first stage retrieval for the web domain [KOM+20, HLY+21, **?**]. However in the web domain we are in a setting with large amounts of training data [NRS+16] and a query-to-passage or a query-to-document retrieval task [CMYC19]. In this study we investigate neural retrieval models in the context of legal case retrieval and prior art search, which are both document-to-document retrieval tasks.

We investigate dense document-to-document retrieval with limited labelled target data for training, in particular legal case retrieval. In order to use DPR models for document-to-document retrieval, we propose a Paragraph Aggregation Retrieval Model (PARM) which liberates DPR models from their limited input length. PARM retrieves documents on the paragraph-level: for each query paragraph, relevant documents are retrieved based on their paragraphs. Then the relevant results per query paragraph are aggregated into one ranked list for the whole query document. For the aggregation we propose vector-based aggregation with reciprocal rank fusion (VRRF) weighting, which combines the advantages of rank-based aggregation [CCB09] and topical aggregation based on the dense embeddings. Experimental results show that VRRF outperforms rank-based aggregation strategies for dense document-to-document retrieval with PARM for legal case retrieval, but not for our test collection of prior art search in the patent domain. We compare PARM to document-level retrieval and demonstrate higher retrieval effectiveness of PARM for lexical and dense first-stage retrieval on two different legal case retrieval collections. We investigate how to train the dense retrieval model for PARM on limited target data with labels on the paragraph or the document-level. In addition, we analyze the differences of the retrieved results of lexical and dense retrieval with PARM.

4.2.1 Introduction

Neural first stage retrieval models, also referred to as dense passage retrieval (DPR) [KOM+20], brought substantial effectiveness gains to information retrieval (IR) tasks in the web domain [GDFC20, KOM+20, XXL+21]. The promise of DPR models is to boost the recall of first stage retrieval by leveraging the semantic information for retrieval as opposed to traditional retrieval models [RZ09], which rely on lexical matching. The web domain is a setting with query-to-passage or query-to-document retrieval tasks and a large amount of training data [CMYC20, NRS+16], while training data is much more limited in other domains [RKG+20, PLHZ11]. Furthermore we see recent advances in neural retrieval remain neglected for document-

to-document retrieval despite the task's importance in several, mainly professional, domains [Loc17, Pir10, RKG+20, RAHK20].

In this work we investigate the effectiveness of dense retrieval models for document-to-document tasks, in particular legal case retrieval and prior art search in the patent domain. We focus on first stage retrieval with dense models and therefore aim for a high recall. The first challenge for DPR models in document-to-document retrieval tasks is the input length of the query documents and of the documents in the corpus. In legal case retrieval and prior art search the cases tend to be long documents [vOS17] with an average length of 1269 words in the COLIEE case law corpus [RKG+20]. However the input length of DPR models is limited to 512 tokens [KOM+20] and theoretically bound of how much information of a long text can be compressed into a single vector [LETC21]. Furthermore we reason in accordance with the literature [BK08, SML+20, WML+20, WML+19] that relevance between two documents is not only determined by the complete text of the documents, but that a candidate document can be relevant to a query document based on one paragraph that is relevant to one paragraph of the query document. In the web domain DPR models are trained on up to $500k$ training samples [NRS+16], whereas in most domain-specific collections only a limited amount of hundreds of labelled samples is available [GNS+19, HTBO09b, RKG+20].

In this work we address these challenges by proposing a **paragraph aggregation retrieval model (PARM)** for dense document-to-document retrieval. PARM liberates dense passage retrieval models from their limited input length without increasing the computational cost. Furthermore PARM gives insight on which paragraphs the document-level relevance is based, which is beneficial for understanding and explaining the retrieved results. With PARM the documents are retrieved on the paragraph-level: the query document and the documents in the corpus are split up into their paragraphs and for each query paragraph a ranked list of relevant documents based on their paragraphs is retrieved. The ranked lists of documents per query paragraph need to be aggregated into one ranked list for the whole query document. As PARM provides the dense vectors of each paragraph, we propose **vector-based aggregation with reciprocal rank fusion weighting (VRRF)** for PARM. VRRF combines the merits of rank-based aggregation [CCB09, dHSMM15] with semantic aggregation with dense embeddings. We investigate:

RQ1.2.1 *How does VRRF compare to other aggregation strategies within PARM?*

We find that our proposed aggregation strategy of VRRF for PARM leads to the highest retrieval effectiveness in terms of recall compared to rank-based [CCB09, SF94] and vector-based aggregation baselines [LYM+20] for legal case retrieval. For prior art search the neural retrieval models either utilized with the PARM architecture or off-the-shelf dense retrieval do not outperform the statistical retrieval model BM25. Furthermore we investigate:

RQ1.2.2 *How effective is PARM with VRRF for document-to-document retrieval?*

We compare PARM with VRRF to document-level retrieval for lexical and dense retrieval methods on two different test collections for the document-to-document task of legal case retrieval. We demonstrate that PARM consistently improves the first stage retrieval recall for dense document-to-document retrieval. Furthermore, dense document-to-document retrieval with PARM and

VRRF aggregation outperforms lexical retrieval methods in terms of recall at higher cut-off values.

The success of DPR relies on the size of labelled training data. As we have a limited amount of labelled data as well as paragraph and document-level labels we investigate:

RQ1.2.3 *How can we train neural retrieval models for PARM for document-to-document retrieval most effectively?*

For training the neural retrieval model DPR for PARM we compare training with relevance labels on the paragraph or document-level for legal case retrieval. We find that despite the larger size of document-level labelled datasets, the additional training data is not always beneficial compared to training DPR on smaller, but more accurate paragraph-level samples. Our contributions are:

- We propose a **paragraph aggregation retrieval model (PARM)** for dense document-to-document retrieval and demonstrate higher retrieval effectiveness for dense retrieval with PARM compared to retrieval without PARM and to lexical retrieval with PARM for the task of prior case retrieval in the legal domain
- We investigate PARM for the task of prior art search in the patent domain and find that dense retrieval models are not beneficial yet for this task, both with PARM and with off-the-shelf dense passage retrieval
- We propose **vector-based aggregation with reciprocal rank fusion weighting (VRRF)** for dense retrieval with PARM and find that VRRF leads to the highest recall for PARM compared to other aggregation strategies.
- We investigate training DPR for PARM and compare the impact of fewer, more accurate paragraph-level labels to more, potentially noisy document-level labels for prior case retrieval.
- We publish the code int the Github repository linked in Section 1.3

4.2.2 Paragraph aggregation retrieval model (PARM)

In this section we propose PARM as well as the aggregation strategy VRRF for PARM for dense document-to-document retrieval and training strategies.

Workflow

We use the DPR model [KOM+20] as described in section 2.6.3.

As the input length of BERT [DCLT19] is limited to 512 tokens, the input length for the query and the candidate passage for DPR [KOM+20] is also limited by that. The length of query and candidate documents for document-to-document tasks exceeds this input length. For example the average length of a document is 1296 words for the legal case retrieval collection COLIEE [RKG+20]. We reason that for document-to-document tasks a single paragraph or multiple paragraphs can be decisive for the relevance of a document to another one [BK08, LC02, WML+20, WML+19]

and that different paragraphs contain different topics of a document. Therefore we propose a **paragraph aggregation retrieval model (PARM)**, in order to use DPR models for dense document-to-document retrieval. PARM retrieves relevant documents based on the paragraph-level relevance.

The workflow of PARM is visualized in Fig. 4.3. For the documents in the corpus we split each document d into paragraphs $p_1, .., p_{m_d}$ with m_d the number of paragraphs of document d. We take the paragraphs of the document as passages for DPR. We index each paragraph p_j, $j \in 1, .., m_d$ of each document d in the corpus and attain a paragraph-level index containing the encoded paragraphs $\hat{p}_j$ for all documents d in the corpus. At query time, the query document q is also split up into paragraphs $q_1, ..., q_{n_q}$, where n_q is the number of paragraphs of q. For each query paragraph q_i with $i \in 1, .., n_q$ the top N most relevant paragraphs are retrieved from the paragraph-level corpus. The result is a ranked list r_i with $i \in 1, .., n_q$ per query paragraph q_i with N relevant paragraphs. The paragraphs in the ranked lists r_i with $i \in 1, .., n_q$ are replaced by the documents that contain the paragraphs. Therefore it is possible that one document occurs multiple times in the list. In order to attain one ranked list for the whole query document q, the ranked paragraph lists of retrieved documents $r_1, ..., r_{n_q}$ of each query paragraph q_i with $i \in 1, .., n_q$ need to be aggregated to one ranked list.

Vector-based aggregation with reciprocal rank fusion weighting (VRRF)

Multiple works have demonstrated the benefit of reciprocal rank fusion [CCB09, dHSMM15, MMM15] for rank-based aggregation of multiple ranked retrieved lists. Using dense retrieval with PARM we have more information than the ranks and scores of the retrieved paragraphs: we have dense embeddings, which encode the semantic meaning of the paragraphs, for each query paragraph and the retrieved paragraphs. In order to make use of this additional information for aggregation, we propose **vector-based aggregation with reciprocal rank fusion weighting (VRRF)**, which extends reciprocal rank fusion for neural retrieval. VRRF combines the advantages of reciprocal rank fusion with relevance signals of semantic aggregation using the dense vector embeddings.

In **VRRF** we aggregate documents using the dense embeddings $\hat{p}_i$ of the passages p_i, which are from the same document d and which are in the retrieved list r_i with $i \in 1, .., n_q$, with a weighted sum, taking the reciprocal rank fusion score [CCB09] as weight. The dense embeddings $\hat{q}_i$ of

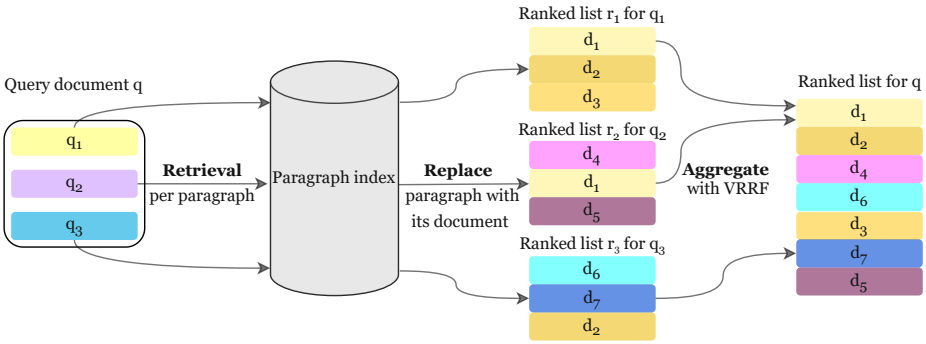


Figure 4.3: PARM workflow for query document q and retrieved documents $d_1, .., d_7$

each query paragraph q_i with $i \in 1, .., n_q$ are aggregated by adding the embeddings without a weighting:

$$\hat{q} = \sum_{i=1}^{n_q} \hat{q}_i \qquad \hat{d} = \sum_{i=1}^{n_q} \sum_{p \in d, d \in r_i} rrf(q_i, p_i)\, \hat{p}_i$$

We compute the relevance score between query and candidate document with the dot-product between the aggregated embedding of query $\hat{q}$ and candidate document $\hat{d}$.

To confirm the viability of VRFF aggregation, we propose simple baselines: **VRanks** and **VScores**, where the paragraph embeddings $\hat{p_i}$ of d are aggregated with the rank or the score of the passage p_i as weight.

Training strategies

As we have a limited amount of labelled target data, we examine how to effectively train a DPR model for PARM with the training collections at hand. We assume that we have test collections consisting of documents with clearly identifiable paragraphs, with relevance assessments on either the paragraph or the document-level.

Paragraph-level training

For the paragraph-level labelled training we take the relevant paragraphs in the training set as positives and sample random negatives from the paragraphs in the corpus. Here we sample as many negatives as we have positive samples for each query paragraph, thereby balancing the training data following a standard methodology when training ranking and retrieval models on triples [SML$^+$20, HLY$^+$21, NC19].

Document-level training

For the document-level labelled training the collection contains query documents and a corpus of documents with relevance assessments for each query document. We sample negative documents randomly from the corpus. In order to use the document-level labelled collection for training the DPR model, we split up the query document as well as the positive documents into its paragraphs and consider each paragraph of the query document relevant to each paragraph of each positive document. Equivalently we consider each paragraph of a negative document irrelevant to each query paragraph. As on average each document in the COLIEE dataset [RKG$^+$20] contains 42.44 paragraphs, one relevant document leads to $42 \cdot 20 = 840$ paragraph-level labels containing one positive and one negative sample to a query paragraph. Therefore this method greatly increases the number of paragraph-level annotations, however this comes with the risk of potentially noisy labels [AYYZL19].

4.2.3 Experiment Design

We introduce the data collections for training and testing the retrieval models and give details about the training and retrieval.

Training and test collections

We focus on the document-to-document task of legal case retrieval and the prior art search in the patent domain because of the importance of the tasks in the respective domains [LZ18, Loc17, SLM+20, SML+20, PLHZ11] which facilitates the availability of training collections with relevance annotations on the paragraph and the document-level [RKG+20]. For training the DPR models, we use paragraph and document-level labelled collections. For the evaluation we use the document-level collections.

Paragraph-level labelled collections

COLIEE [RKG+20] is a competition for legal information extraction and retrieval which provides datasets for legal case retrieval and case entailment as described in section 2.5. Task 2 of COLIEE 2020 [RKG+20] provides a training and test collection for legal case entailment. It contains relevance labels on the legal case paragraph level, given a query claim, a set of candidate claims to the query claim as well as relevance labels for the candidate claims. We denote these sets with *COLIEEPara train/test*.

For prior art search we use the training and test datasets from the claims to passage retrieval task from the CLEF-IP evaluation campaign, which we introduced in section 2.5. The task provides a training and test collection based on claims citing relevant passages from other patent documents. We denote the sets with *CLEFIPPara train/test*.

Document-level labelled collections

In Task 1 of COLIEE 2021 [RKG+20], the legal case retrieval task, query cases with their relevance judgements on the document-level are provided together with a corpus of candidate documents. The corpus consists of 4415 legal cases and a set of training and test queries with relevance annotations is given. The relevant cases are the cases which are referenced in the query case, the references are not on a paragraph-level but on the document-level. We divide the training set of COLIEEDoc into a training and validation set. The validation set contains the last 100 queries of the training set from query case 550 to 650. We will denote the training, validation and test collection with *COLIEEDoc train/val/test*.

For a broader evaluation, we evaluate our models additionally on the CaseLaw collection [Loc17] introduced in section 2.5. It contains a corpus of legal cases, query cases and their relevance judgements for legal case retrieval.

For the task of prior art search in the patent domain, we use the CLEF-IP prior art search training and test datasets, introduced in section 2.5. The document-level training set is divided in a training and validation set. We denote the training and test collection on the document-level with *CLEFIPDoc train/val/test*. The statistics for the sets can be found in Table 4.4.

Data preprocessing

For COLIEEDoc, we remove the French versions of the cases, we divide the cases into an introductory part, a summary, if it contains one, and its claims, which are indicated by their numbering. As indicated in Table 4.4, the paragraphs have an average length of 84 words and 96.2% of the paragraphs are not longer than 512 words. The documents in the corpus have an

Table 4.4: Statistics of paragraph- and document-level labelled collections.

Labels	Dataset	Train/ Test	Statistics					
			# queries	∅ # docs	∅ # rel docs	∅ para length	% para < 512 words	∅ # para
Para	COLIEEPara	Train	325	32.12	1.12	102	95.5%	-
	COLIEEPara	Test	100	32.19	1.02	117	95.2%	-
	CLEFIPPara	Train	44	3.5M	8.5	115	96.1%	-
	CLEFIPPara	Test	42	3.5M	18.2	92	97.8%	-
Doc	COLIEEDoc	Train	650	4415	5.17	84	96.2%	44.6
	COLIEEDoc	Test	250	4415	3.60	92	97.8%	47.5
	CaseLaw	Test	100	63431	7.2	219	91.3%	7.5
	CLEFIPDoc	Train	351	3.5M	19.5	43.4	98.8%	25.9
	CLEFIPDoc	Test	100	3.5M	19.4	43.6	98.6%	25.7

average length of 3309 words and contain on average 20 paragraphs with an average length of 164 words. The query descriptions in the AILA dataset have an average length of 454 words, therefore we split them up into paragraphs with an average length 222 words to fit the input size of the DPR model and get on average 2.04 paragraphs per query. The CaseLaw dataset is split along the line breaks of the text and merged to paragraphs by concatenating sentences until the paragraphs exceed the length of 200 words. The documents in the corpus have an average length of 1669 words, the query cases have an average length of 6341 words and the paragraphs 219 words. The documents in the corpus contain on average 8 paragraphs, the query cases 31 paragraphs.

For CLEF-IP we filter out the non-English patents and only use English patents in the training and test collection, both for the paragraph-level and the document-level sets. For CLEFIPDoc, the patent documents have a title, abstract, claims and a technical description. Preliminary experiments showed that concatenating the claims to one paragraph achieves the highest effectiveness for retrieval with PARM BM25. Thus we leverage PARM with 3 paragraphs: the title and abstract, all concatenated claims, and the description.

Baselines

As baseline we use the statistical retrieval model BM25 [RZ09], as described in section 2.6.1. For BM25 we use ElasticSearch[6] with parameters $k = 1.3$ and $b = 0.8$, which we optimized on COLIEEDocval.

VRRF aggregation for PARM (RQ1.2.1)

In order to investigate the retrieval effectiveness of our proposed aggregation strategy VRFF for PARM, we compare VRRF to the commonly used score-based aggregation strategy CombSum

[6] https://github.com/elastic/elasticsearch

[SF94] and rank-based aggregation strategy of reciprocal rank fusion (RRF) [CCB09] for PARM. Since PARM is not effective for the CLEF-IP test collection, we only investigate the research question for legal case retrieval.

As baselines for vector-based aggregation, we investigate VSum, VMin, VMax, VAvg, which are originally proposed by Li et al. [LYM+20] for re-ranking on a passage-to-document retrieval task. In order to use VSum, VMin, VMax, VAvg in the context of PARM, we aggregate independently the embeddings of both, the query and the candidate document. In contrast to Li et al. [LYM+20] we aggregate the query and paragraph embeddings independently and score the relevance between aggregated query and aggregated candidate embedding after aggregation. The learned aggregation methods of CNN and Transformer proposed by Liu et al. [LYM+20] are therefore not applicable to PARM, as they learn a classification on the embedding of the concatenated query and paragraph.

PARM VRRF for dense document-to-document retrieval (RQ1.2.2)

In order to investigate the retrieval effectiveness of PARM with VRRF for dense document-to-document retrieval, we compare PARM to document-level retrieval on two document-level collections (COLIEEDoc and CaseLaw) for legal case retrieval and on CLEFIPDoc for prior art search. Because of the limited input length, the document-level retrieval either reduces to retrieval based on the First Passage (FirstP) or the passage of the document with the maximum score (MaxP) [AYWY+19, ZYL21]. In order to separate the impact of PARM for lexical and dense retrieval methods, we also use PARM with BM25 as baseline. For PARM with BM25 we also investigate which aggregation strategy leads to the highest retrieval effectiveness in order to have a strong baseline. As BM25 does not provide dense embeddings only rank-based aggregation strategies are applicable.

Paragraph and document-level labelled training (RQ1.2.3) We train a DPR model on a paragraph- and another document-level labelled collection and compare the retrieval performance of PARM for document-to-document retrieval. As bi-encoders for DPR we choose BERT [DCLT19] and LegalBERT [CFM+20] and SciBERT [BLC19].

For legal case retrieval, we train DPR on the paragraph-level labelled collection COLIEEPara train and additionally on the document-level labelled collection COLIEEDoc train as described in Section 4.2.2. For prior art search, we train DPR respectively on the paragraph-level and document-level training sets of CLEFIPPara and CLEFIPDoc. We use the public code[7] and train DPR according to Karpukhin et al. [KOM+20]. We sample the negative paragraphs randomly from randomly sampled negative documents and take the 20 paragraphs of a positive document as positive samples, which have the highest BM25 score to the query paragraph. This training procedure lead to the highest recall compared to training with all positive paragraphs or with BM25 sampled negative paragraphs. We also experimented with the DPR model pre-trained on open-domain QA as well as TAS-balanced DPR model [HLY+21], but initial experiments did not show a performance improvement. We train each DPR model for 40 epochs and take the best checkpoint according to COLIEEPara test/COLIEEDoc val. We use batch size of 22 and a learning rate of $2 * 10^{-5}$, after comparing three commonly used learning rates ($2 * 10^{-5}$, $1 * 10^{-5}$, $5 * 10^{-6}$) for [KOM+20].

[7] https://github.com/facebookresearch/DPR

Table 4.5: Aggregation comparison for PARM on COLIEEval, VRRF shows best results for dense retrieval, stat. sig. difference to RRF w/ paired t-test ($p<0.05$) denoted with †, Bonferroni correction with n=7. For BM25 only rank-based methods applicable.

Aggregation	BM25			DPR BERT			DPR LegalBERT		
	R@100	R@500	R@1K	R@100	R@500	R@1K	R@100	R@500	R@1K
Rank-based									
CombSum [SF94]	.5236	.7854	.8695	.4460	.7642	.8594	.5176	.7975	.8882
RRF [CCB09]	**.5796**	**.8234**	**.8963**	.5011	.8029	.8804	**.5830**	.8373	.9049
Vector-based									
VAvg [LYM+20]	-	-	-	.1908†	.4668†	.6419†	.2864†	.4009†	.7466†
VMax [LYM+20]	-	-	-	.3675†	.6992†	.8273†	.4071†	.6587†	.8418†
VMin [LYM+20]	-	-	-	.3868†	.6869†	.8295†	.4154†	.6423†	.8465†
VSum [LYM+20]	-	-	-	.4807	.7496†	.8742	.5182†	.8069	.8882
Vector-based with rank-based weights (Ours)									
VScores	-	-	-	.4841	.7616†	.8709	.5195†	.8075†	.8882†
VRanks	-	-	-	.4826	.7700†	.8804	.5691†	.8212	.8980
VRRF	-	-	-	**.5035**	**.8062**†	**.8806**	**.5830**†	**.8386**†	**.9091**†

4.2.4 Results and Analysis

We evaluate the first stage retrieval performance with nDCG@10, recall@100, recall@500 and recall@1k using pytrec_eval. We focus our evaluation on recall because the recall performance of the first stage retrieval bounds the ranking performance after re-ranking the results in the second stage for a higher precision. We do not compare our results to the reported state-of-the-art results as they rely on re-ranked results and do not report evaluation results after the first stage retrieval.

RQ1.2.1: VRRF aggregation for PARM

As we propose vector-based aggregation with reciprocal rank fusion weighting (VRRF) for PARM, we first investigate:
(**RQ1.2.1**) *How does VRRF compare to other aggregation strategies within PARM?*
We compare VRRF, which combines dense-vector-based aggregation with rank-based weighting, to score/rank-based and vector-based aggregation methods for PARM. The results in Table 4.5 show that VRRF outperforms all rank and vector-based aggregation approaches for the dense retrieval results of DPR PARM with BERT and LegalBERT. For the lexical retrieval BM25 with PARM, only rank-based aggregation approaches are feasible, here RRF shows the best performance, which will be our baseline for RQ1.2.2.

RQ1.2.2: PARM VRRF vs Document-level retrieval

As we propose PARM VRRF for document-to-document retrieval, we investigate:
(**RQ1.2.2**) *How effective is PARM with VRRF for document-to-document retrieval?*
We evaluate and compare PARM and document-level retrieval for lexical and dense retrieval

Table 4.6: Doc-to-doc retrieval results for PARM and Document-level retrieval for legal case retrieval on COLIEEDoc and CaseLaw. No comparison to results reported in prior work as those rely on re-ranking, while we evaluate only first stage retrieval evaluation. *nDCG cutoff at 10, stat. sig. difference to BM25 Doc w/ paired t-test ($p < 0.05$) denoted with † and Bonferroni correction with n=12, effect size >0.2 denoted with ‡.*

Model	Retrieval	COLIEEDoc test				CaseLaw			
		nDCG	R@100	R@500	R@1K	nDCG	R@100	R@500	R@1K
BM25									
BM25	Doc	**.2435**	.6231	.7815	.8426	**.2653**	.4218	.5058	.5438
	PARM RRF	.1641†‡	**.6497**†‡	.8409†‡	.8944†‡	.0588†‡	.3362†‡	.5716†‡	.6378†‡
DPR									
BERT para	Doc FirstP	.0427†‡	.3000†‡	.5371†‡	.6598†‡	.0287†‡	.0871†‡	.1658†‡	.2300†‡
	Doc MaxP	.0134†‡	.1246†‡	.5134†‡	.6201†‡	.0000†‡	.0050†‡	.4813†‡	.4832†‡
	PARM RRF	.0934†‡	.5765†‡	.8153†‡	.8897†‡	.0046†‡	.1720†‡	.5019†‡	.5563†
	PARM VRRF	.0952†‡	.5786†‡	.8132†‡	.8909†‡	.1754†‡	.3855†‡	.5328†‡	.5742†‡
LegalBERT para	Doc FirstP	.0553†‡	.2447†‡	.4598†‡	.5657†‡	.0397†‡	.0870†‡	.1844†‡	.2248†‡
	Doc MaxP	.0073†‡	.0737†‡	.3970†‡	.5670†‡	.0000†‡	.0050†‡	.4846†‡	.4858†‡
	PARM RRF	.1280†‡	.6370	.8308†‡	.8997†‡	.0177†‡	.2595†‡	.5446†‡	.6040†‡
	PARM VRRF	.1280†‡	.6396	.8310†‡	.9023†‡	.0113†‡	**.4986**†‡	.5736†‡	.6340†‡
LegalBERT doc	Doc FirstP	.0682†‡	.3881†‡	.6187†‡	.7361†‡	.0061†‡	.0050†‡	.4833†‡	.4866†‡
	Doc MaxP	.0008†‡	.0302†‡	.2069†‡	.2534†‡	.0022†‡	.0050†‡	.4800†‡	.4833†‡
	PARM RRF	.1248†‡	.6086†	.8394†‡	.9114†‡	.0117†‡	.2277†‡	.5637†‡	.6265†‡
	PARM VRRF	.1256†‡	.6127†	**.8426**†‡	**.9128**†‡	.2284†‡	.4620†‡	**.5847**†‡	**.6402**†‡

methods on the two test collections (COLIEEDoc and CaseLaw) for document-to-document retrieval in Table 4.6. For prior art search we compare PARM to document-level retrieval for lexical and dense retrieval models on CLEF-IPDoc in Table **??**. For legal case retrieval, for BM25 we find that PARM-based retrieval outperforms document-level retrieval at each recall stage, except for R@100 on CaseLaw.

For dense retrieval we evaluate DPR models with BERT trained solely on the paragraph-level labels and with LegalBERT trained on the paragraph-level labels (denoted with LegalBERT para) and with additional training on the document-level labels (denoted with LegalBERT doc). For dense document-to-document retrieval PARM consistently outperforms document-level retrieval for all performance metrics for both test collections. Furthermore PARM aggregation with VRRF outperforms PARM RRF in nearly all cases. Overall we find that LegalBERTdoc-based dense retrieval with PARM VRRF achieves the highest recall at high ranks. When comparing the nDCG@10 evaluation we find that PARM lowers the nDCG@10 score for BM25 as well as for dense retrieval. Therefore we suggest that PARM is beneficial for first stage retrieval, so that in the re-ranking stage the overall ranking can be improved.

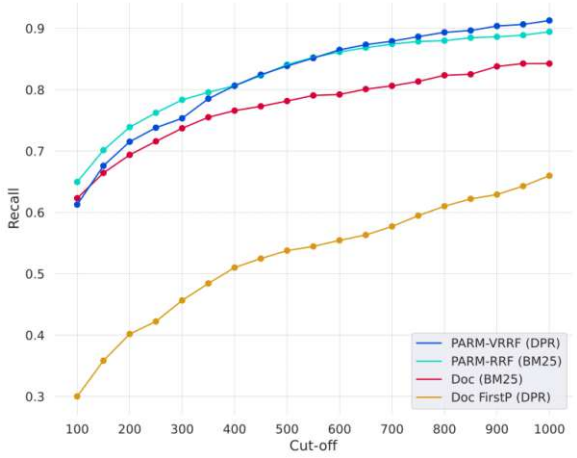


Figure 4.4: Recall at different cut-off values for PARM-VRRF (DPR) and PARM-RRF (BM25) and Document-level retrieval with BM25 and DPR for COLIEEDoc test.

	COLIEEDoc		CaseLaw	
	BM25	DPR	BM25	DPR
Total				
relevant	900	900	720	720
PARM	892	896	578	545
Doc	751	662	419	199
Sets				
PARM ∩ Doc	750	661	417	196
PARM\Doc	142	235	161	349
Doc\PARM	1	1	2	3

Figure 4.5: Number of relevant documents retrieved in comparison between PARM and Doc-level retrieval for COLIEEDoc and CaseLaw with BM25 or LegalBERT_doc-based DPR.

For legal case retrieval we further analyze the results. In Figure 4.4 we show the recall at different cut-off values for PARM-VRRF with DPR (based on LegalBERTdoc) and PARM-RRRF with BM25 compared to document-level retrieval (Doc FirstP) of BM25/DPR. When comparing PARM to document-retrieval, we can see a clear gap between the performance of document-level retrieval and PARM for BM25 and for DPR. Furthermore we see that dense retrieval (PARM-VRRF DPR) outperforms lexical retrieval (PARM-RRF BM25) at cut-off values above 500.

In order to analyze the differences between PARM and document-level retrieval further, we analyze in Figure 3.5, how many relevant documents are retrieved with PARM or with document-level retrieval with lexical (BM25) or dense methods (DPR). Furthermore we investigate how many relevant documents are retrieved by both PARM and document-level retrieval (PARM ∩ Doc), and how many relevant documents are retrieved only with PARM and not with document-level retrieval (PARM\Doc) and vice versa (Doc\PARM). When comparing the performance of PARM and document-level retrieval, we find that PARM retrieves more relevant documents in total for both test collections. PARM retrieves $142 - 380$ of the relevant documents that did not get retrieved with document-level retrieval (PARM\Doc), which are $15 - 52\%$ of the total number of relevant documents. This analysis demonstrates that PARM largely retrieves many of relevant documents that are not retrieved with document-level retrieval. We conclude that PARM is not only beneficial for dense but also for lexical retrieval for legal case retrieval.

For prior art search, we see in Table 4.7 that BM25 on the document-level outperforms all neural retrieval approaches, both with PARM and also with the FirstP retrieval. This holds true in terms of nDCG@10 as well as for recall at all cut-offs from 100 to 1000. Thus we do not investigate any other aggregation approaches for PARM like VRRF or document-level retrieval with MaxP.

Overall we find that PARM greatly improves the first stage retrieval for legal case retrieval,

Table 4.7: Doc-to-doc retrieval results for PARM and Document-level retrieval on CLEFIPDoc for prior art search. No comparison to results reported in prior work as those rely on re-ranking, while we evaluate only first stage retrieval evaluation. *nDCG cutoff at 10, stat. sig. difference to BM25 Doc w/ paired t-test ($p < 0.05$) denoted with † and Bonferroni correction with n=12, effect size >0.2 denoted with ‡.*

Model	Retrieval	CLEFIPDoc test					
		nDCG	R@100	R@200	R@300	R@500	R@1k
BM25							
BM25	Doc	**.0562**†‡	**.1369**†‡	**.1620**†‡	**.1746**†‡	**.1989**†‡	**.2405**†‡
	PARM RRF	.0344	.1043	.1405	.1539	.1764	.2144
DPR							
BERT	Doc FirstP	.0022	.0224	.0362	.0479	.0673	.0801
	PARM RRF	.0059	.0244	.0432	.0521	.0600	.0703
LegalBERT	Doc FirstP	.0040	.0233	.0362	.0489	.0579	.0812
	PARM RRF	.0077	.0240	.0313	.0399	.0528	.0832
SciBERT	Doc FirstP	.0027	.0356	.0404	.0452	.0574	.0803
	PARM RRF	.0055	.0279	.0499	.0586	.0733	.0906

however this does not hold true for the CLEF-IP patent test collection. We suggest that there needs to be further future work to investigate PARM for prior art search, since the CLEF-IP test collection only relies on the citations of the patent examiners and thus the relevance labels are heavily biased towards the retrieval model employed in the search of the patent examiners. Most probably the retrieval system employed in the labelling run is based on keywords and thus benefits BM25. Therefore one future direction could be to analyze the CLEF-IP test collection for its suitability to evaluate neural first stage retrieval models.

RQ1.2.3: Paragraph-level vs Document-level Labelled Training

As labelled in-domain data for document-to-document retrieval tasks is limited, we ask: (**RQ1.2.3**) *How can we train neural retrieval models for PARM for document-to-document retrieval most effectively?* We compare the retrieval performance for BERT-based and LegalBERT-based dense retrieval models in Table 4.8 for legal case retrieval, which are either trained solely on the paragraph-level labelled collection or additionally trained on the document-level labelled collection. The upper part of the table shows that for BERT the additional training data on document-level improves the retrieval performance for document-level retrieval, but harms the performance for PARM RRF and PARM VRRF. For LegalBERT the additional document-level training data highly improves the performance of document-retrieval. For PARM the recall is improved at higher cut-off values (@500, @1000) for a cut-off. Therefore we consider the training on document-level labelled data beneficial for dense retrieval based on LegalBERT. This reveals that it is not always better to have more, potentially noisy data, for BERT-based dense retrieval the training with fewer, but accurate paragraph-level labels is more beneficial for overall

Table 4.8: Paragraph- and document-level labelled training of DPR. Document-level labelled training improves performance at high ranks for LegalBERT, statistical significantly different to paragraph-level training compared to paragraph- and document-level training with paired t-test ($p < 0.05$) denoted with † (Comparison for each model is training with para Labels vs training with para+doc Labels)

Model	Retrieval	Train Labels	COLIEEDoc val R@100	R@200	R@300	R@500	R@1K
DPR Retrieval							
BERT	Doc FirstP	para	.3000	.4018	.4566	.5371	.6598
	Doc FirstP	+ doc	.3800†	.4641†	.5160†	.6054†	.7211†
	PARM RRF	para	.5765	.6879	.7455	.8153	.8897
	PARM RRF	+ doc	.5208†	.6502†	.7100†	.7726†	.8660
	PARM VRRF	para	.5786	.6868	.7505	.8132	.8909
	PARM VRRF	+ doc	.5581†	.6696	.7298†	.7970	.8768
LegalBERT	Doc FirstP	para	.2447	.3286	.3853	.4598	.5657
	Doc FirstP	+ doc	.3881†	.4665†	.5373†	.6187†	.7361†
	PARM RRF	para	.6350	.7323	.7834	.8308	.8997
	PARM RRF	+ doc	.6086†	.7164	.7561†	.8394	.9114
	PARM VRRF	para	**.6396**	**.7325**	**.7864**	.8310	.9023
	PARM VRRF	+ doc	.6098†	.7152	.7520†	**.8396**	.9128†

document-to-document retrieval with PARM.

Analysis of paragraph relations

With our proposed paragraph aggregation retrieval model for dense document-to-document retrieval we can analyze on which paragraphs the document-level relevance is based. To gain more insight in what the dense retrieval model learned to retrieve on the paragraph-level with PARM, we analyze which query paragraph retrieves which paragraphs from relevant documents with dense retrieval with PARM and compare it to lexical retrieval with PARM. In Figure 4.6, a heatmap visualizes which query paragraph how often retrieves which paragraph from a relevant document with PARM BM25 or PARM DPR on the COLIEEDoc test set. As introduced in Section 4.2.3, the legal cases in COLIEEDoc contain an introduction, a summary and claims as paragraphs. For the introduction (I) and the summary (S) we see the paragraph relation for lexical and dense retrieval that both methods retrieve also more introductions and summaries from the relevant documents. We reason this is due to the special structure of the introduction and the summary which is distinct to the claims. For the query paragraphs 1.-10. we see that PARM DPR seems to focus on to the diagonal different to PARM BM25. This means for example that the first paragraph retrieves more first paragraphs from relevant documents than they retrieve other paragraphs. As the claim numbers are removed in the data preprocessing, this focus relies on the textual content of the claims. This paragraph relation suggests that there is a topical or

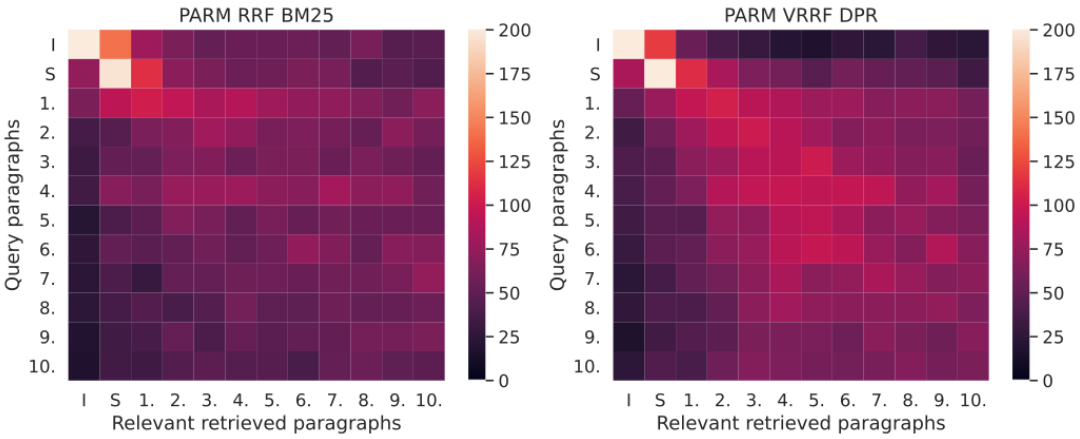


Figure 4.6: Heatmap for PARM retrieval with BM25 or DPR visualizing which query paragraph how often retrieves which paragraph from a relevant document. I denotes the introduction, S the summary, 1.-10. denote the claims 1.-10. of COLIEEDoc test.

hierarchical structure in the claims of legal cases, which is learned by DPR and exhibited with PARM. This structural component can not be exhibited with document-level retrieval.

4.2.5 Conclusion

In this work we address the challenges of using dense passage retrieval (DPR) in first stage retrieval for document-to-document tasks with limited labelled data. We propose the paragraph aggregation retrieval model (PARM), which liberates dense passage retrieval models from their limited input length and which takes the paragraph-level relevance for document retrieval into account. We demonstrate for legal case retrieval on two test collections higher first stage recall for dense document-to-document retrieval with PARM than with document-level retrieval. For legal case retrieval, we also show that dense retrieval with PARM outperforms lexical retrieval with BM25 in terms of recall at higher cut-off values. As part of PARM we propose the novel vector-based aggregation with reciprocal rank fusion weighting (VRFF), which combines the advantages of rank-based aggregation with RRF [CCB09] and topical aggregation with dense embeddings. We demonstrate the highest retrieval effectiveness for PARM with VRRF aggregation compared to rank and vector-based aggregation baselines. For the document-to-document retrieval task of prior art search, we find that dense retrieval approaches based on PARM and off-the-shelf dense retrieval do not outperform lexical retrieval with BM25. Furthermore we investigate how to train dense retrieval models for dense document-to-document retrieval with PARM. For legal case retrieval, we find the interesting result that training DPR models on more, but noisy document-level data does not always lead to overall higher retrieval performance compared to training on less, but more accurate paragraph-level labelled data. Finally, we analyze how PARM retrieves relevant paragraphs and find that the dense retrieval model learns a structural paragraph relation which it exhibits with PARM and therefore benefits the retrieval effectiveness.

Limitations and future work

Similarly to the limitations described in section 4.1.5 this study also is limited by its focus on the English language and future work is needed to investigate if the findings of the study also generalize to other languages. Additionally this work is also limited to the language model BERT as encoder and future work needs to determine how the choice of large language model for the DPR model influences the effectiveness of PARM.

Another limitation is the generalizability of the findings regarding PARM for document-to-document retrieval tasks with documents, which do not have a predefined structure as the legal cases. For the legal cases, the predefined claim structure captures topically coherent blocks of the document and since single claims are relevant to other claims of other document, PARM uses this characteristic of the legal case retrieval task to its advantage. It remains unclear and needs future work, how the findings of PARM generalize to document-to-document retrieval tasks, where the splitting up in paragraphs is not predefined by semantically coherent paragraphs in a structured document.

Another limitation of this study is the reliability of the CLEF-IP test collection for evaluating neural first stage retrieval models that have not participated in the pool creation for annotation. In order to have confidence in the evaluation results of the CLEF-IP test collection, we see some future work in analyzing the suitability of the CLEF-IP test collection to evaluate neural first stage retrieval models since only statistical models contributed to the pool for creating the test collection.

CHAPTER 5

Addressing Data Availability for Evaluation and Training

After investigating domain-specific neural ranking and retrieval models for the document-to-document retrieval tasks of prior case retrieval and prior art search, we now want to focus on the problem of data availability for domain-specific retrieval for evaluation and for training data and investigate the research question:

RQ2 How can the problem of limited available annotated evaluation and training data be addressed in domain-specific retrieval?

We divide this chapter in two sections. In the first section we describe our annotation campaign for creating a human-annotated health test set and compare it to relevance signals from user clicks. In the second section we investigate how we can train neural ranking and retrieval models under a limited annotation and training budget. Here we investigate two scenarios: in the first scenario we train a neural ranking and retrieval model for web search, in the second scenario we adapt a fine-tuned neural ranker to the health domain. Since we want to disentangle the effects of active learning for efficient training data annotation from a possible impact of training domain-specific neural ranking and retrieval architectures like BERT-PLI [SML+20] or PARM, we investigate these active learning strategies for "common" neural ranking and retrieval model architectures [NC19, KOM+20, KZ20], that do not consider long documents. Thus in this chapter we consider ad-hoc retrieval tasks in the web and health domain, that do not contain long documents as queries or in the collection and thus the neural ranking and retrieval models introduced in chapter 2.6 do not need to be adapted in their architecture for these tasks. The second reason for choosing the tasks of ad-hoc retrieval in the web domain and the medical domain when investigating active learning methods for data efficient training of neural rankers, is the availability of large-scale, annotated training data for these two tasks [NRS+16, RLS+21]. Since we do not have the resources to do an interactive annotation and training process with annotators, we simulate the selection and annotation process in the training of the neural rankers and for that simulation we

need large-scale, annotated training datasets, like MS Marco [NRS+16] and TripClick [RLS+21]. Since ad-hoc retrieval tasks are precision-oriented, our evaluation in this chapter is focussed on precision-oriented metrics.

5.1 Capturing Relevance Signals in Annotation

In this chapter we investigate how human-labelled annotations compare to relevance signals from clicks for evaluating domain-specific retrieval in the health domain, answering the research question:

RQ2.1 How do human-label annotations compare to click signals for medical ad-hoc retrieval?

For this investigation we consider the task of medical ad-hoc retrieval, which is exemplified in the **Search Example ❸**. This chapter is based on the publication [AHVH22].

Recently there is a growing interest in evaluating retrieval systems for domain-specific retrieval tasks, however these tasks often lack a reliable test collection with human-annotated relevance assessments following the Cranfield paradigm [RLS+21, RKG+20, PLHZ11, TRR+21]. In the medical domain, the TripClick collection was recently proposed [RLS+21], which contains click log data from the Trip search engine and includes two click-based test sets. However the clicks are biased to the retrieval model used, which remains unknown, and a previous study shows that the test sets have a low judgement coverage for the Top-10 results of lexical and neural retrieval models [HASH22]. We present the novel, relevance judgement test collection TripJudge for TripClick health retrieval. We collect relevance judgements in an annotation campaign and ensure the quality and re-usability of TripJudge by a variety of ranking methods for pool creation, by multiple judgements per query-document pair and by an at least moderate inter-annotator agreement. Since the annotators are students with no particular expertise in health and medicine, we evaluate the agreement between students' annotations with the annotations of experts in the health and medical domain and find a moderate agreement. Furthermore we compare a subset of the non-expert annotations with annotations done by medical experts and find a high overlap between the labels. We compare system evaluation with TripJudge and TripClick and find that that click and judgement-based evaluation can lead to substantially different system rankings.

5.1.1 Introduction

Reliable and robust evaluation of ranking systems is crucial to Information Retrieval (IR) research [Zob98]. Thus a great effort in IR research is put into creating reusable and robust test collections [Voo18, VSL22, Sob17] in task-specific evaluation campaigns like TREC [CCV12, CMY+21b] or CLEF. These campaigns follow the Cranfield paradigm [Cle91] to create relevance judgements on the pooled output of the participating systems. Recently there has been a growing interest in evaluating the retrieval performance of retrieval models for domain-specific retrieval tasks [TRR+21, ZXM+22, HASH22, HKA+22, AHH21] including the medical domain [RLS+21, RDV+17, XLS+20]. Domain-specific retrieval tasks often lack a reliable test collection with

human relevance judgments following the Cranfield paradigm [RLS+21, TRR+21, RKG+20]. Furthermore it remains unclear how well old test collections can be used to evaluate neural retrieval models, which were not part of the pooling process [VSL22].

In the medical domain, a recently proposed benchmark is the publicly available TripClick collection [RLS+21], which we introduced in section 2.5. TripClick contains large-scale click logs from Trip, an English health search engine with professional and non-professional users. It provides two test sets with labels based on the clicks of the users, either estimating relevance by the raw clicks ('Raw') or by the rate of clicks of a document over all retrieved documents for a query ('DCTR'). As the TripClick test sets are based on the clicks of the users, the test sets are biased towards the retrieval model employed by the search engine [Whi13], which remains unknown. In a previous study [HASH22] the test sets were shown to have a low annotation coverage of at most 41% of the Top10 results for lexical and neural retrieval models.

In this work we address these shortcomings of click-based test collections by creating TripJudge, a relevance judgement test collection for TripClick. We collect relevance judgements by running an annotation campaign on the test set queries of TripClick. In order to increase the re-usability of our test collection [BDSV07], we use three participating systems for the pool creation from Hofstätter et al. [HASH22] employing lexical and neural retrieval models. To control the quality of the relevance assessments we monitor the annotation time per query, we employ a graded relevance scheme [GL10, AM12] and we employ multiple relevance assessments per query-document pair (we aim for three assessments but have at least two). We reach an at least moderate inter-annotator agreement measured with Cohen's Kappa.

In order to ensure the quality of non-expert annotations, we conduct an additional annotation campaign with medical experts. In this annotation campaign the health and medical experts annotate a subset of the TripJudge test set using the same set-up as for the annotation campaign with the students. We aggregate the expert annotations with majority vote (the same methodology as for the non-expert annotations) and reach a moderate inter-annotator agreement. We compare the resulting expert annotations with the non-expert annotations and find a high accuracy between both: 78% of binary relevance labels are the same between experts and non-experts.

Furthermore we compare the click and judgement-based evaluation of various retrieval systems and investigate how the rankings of the systems change when evaluated with TripJudge or TripClick. Related work about comparing clicks and judgements for evaluation come to different conclusions. While Joachims et al. [JGP+05] and Zobel et al. [Zob98] find reasonable agreement between the clicks and relevance judgements, Kamps et al. [KKT09] demonstrate that system rank correlations between evaluation based on clicks or judgements is low. First we analyze the overlap of the click-based test collections with TripJudge and find that the majority of the documents that are judged as relevant are not labelled in the click-based collections and therefore are considered irrelevant during evaluation. Similar to related work [Voo98, BV04, VB02], we measure system rank correlation between evaluation with different test collections with Kendalls τ correlation [Ken38]. We find that the rankings with the evaluation of TripJudge differ from the rankings with the click-based test collections. Our contributions are the following:

- We create the relevance judgement-based test collection TripJudge for TripClick health

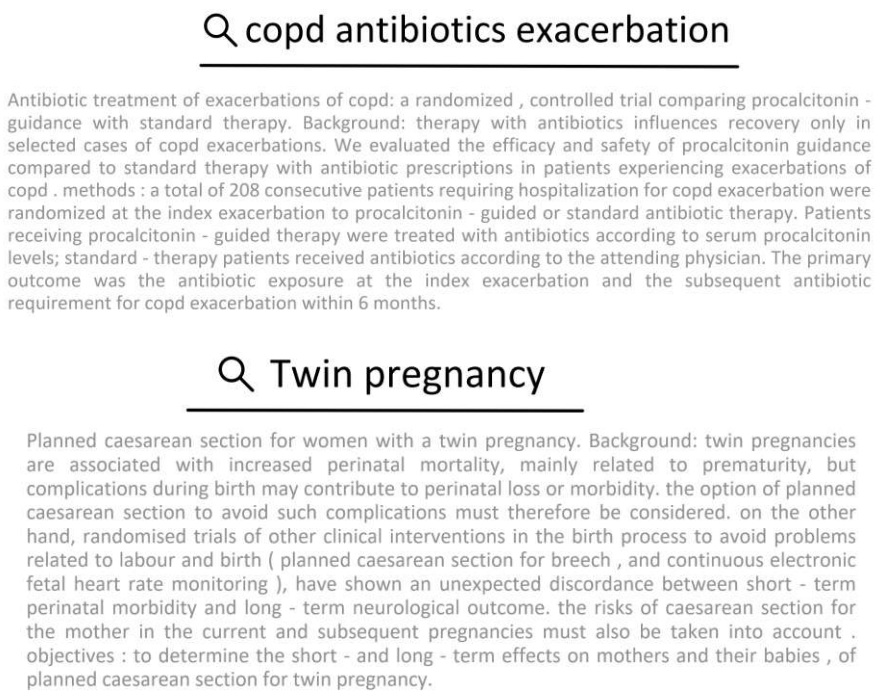


Figure 5.1: Two examples of the TripClick dataset: the query "copd antibiotics exacerbation" and the query "twin pregnancy" with the text of a document, that was clicked by users (labelled as relevant in the TripClick test set) below in light gray.

retrieval and make it publicly available in the Github repository linked in Section 1.3;

- We ensure the quality and re-usability of TripJudge by a variety of systems for pool creation, by multiple judgements per query-document pair, and by an at least moderate inter-annotator agreement in our annotation campaign;
- We compare a subset of the non-expert annotations with annotations of medical expert and find a high accuracy between the labels from the non-experts and experts.
- We compare evaluation with click-based TripClick and our judgment-based TripJudge and find that click and judgment-based evaluation can lead to different system rankings

5.1.2 TripClick dataset

As introduced in chapter 2.5, TripClick is a large-scale dataset of click logs, derived from user engagements on the Trip Database health web search engine. This click log dataset encompasses approximately 5.2 million user interactions and real user queries, systematically collected during the period from 2013 to 2020. An example of a user query and documents that were clicked by the user is visualized in Figure 5.1.

5.1.3 Methodology

We describe how we preprocess the TripClick test queries as well as the pool creation followed by the annotation campaign.

Data and Pool Preparation

In the TripClick test sets the queries are grouped by their user interaction frequency into Head, Torso and Tail. For the annotation campaign we used the 1175 head queries. Head queries are selected because these are the queries that more different users ask in the Trip engine, thus there are more user interaction relevance signals for those queries and the TripClick labels are based on multiple different user interaction signals, thus are more reliable. During the campaign we notice duplicate queries, which differ in their casing (lower/uppercasing) or queries without any natural text (for example "#1 or #2"). We discard these queries from our TripJudge test collection and end up with 1136 unique queries[1].

For the pool creation we use the runs from Hofstätter et al. [HASH22]. In order to have different first stage retrieval methods we use the lexical retrieval run with BM25 [RZ09] (run 1 in Table 5.2) as well as the DPR SciBERT run (run 2 in Table 5.2) which is based on dense retrieval [HLY+21, AHS+22]. As additional run we use the Ensemble which re-ranks BM25 Top-200 candidates using an Ensemble of MonoBERT based on SciBERT, PubMedBERT-Abstract and PubMedBert-Full Text (run 7 in Table 5.2). We create the pool by taking the union of the query-document pairs from the Top-10 of the three runs for all test queries and keep the highest rank among the three runs. This results in a total of 29581 pairs and prioritize the annotation according to the rank of the document: all Top-n pairs have priority $10 - k(k \in [1..n])$; the higher the priority, the earlier they are selected for annotation in the annotation interface.

In order to maintain a low latency in our annotation system, we needed to truncate the documents. As the dense retrieval models rely on the text up to 512 BERT tokens, we truncate the document text to this length, which applies to 10% of the documents.

Annotation Campaign

We conduct the annotation campaign among 135 computer science students with a target of 300 annotations per annotator and reach an average of 287 annotations per annotator (Table 5.1). We aim for a high number of different annotators to not exhaust the annotators and in order to collect a large variety of relevance signals from different annotators. The background of the annotators is that they are computer science students familiar with information retrieval tasks who had a lecture about how to annotate query-document pairs for relevance.

The users of the Trip search engine are a mix of experts and non-experts. As the annotators are non-experts and as previous work points out the challenges with students as annotators [PZB+16, BCS+08], we take several steps to ensure and monitor the quality of the annotations: **1)** We use a 4-graded, ordinal relevance scheme: *Wrong (1)*, *Topic (2)*, *Partial (3)*, and *Perfect (4)*, as suggested in related work [GL10, AM12] **2)** Having more annotators per query-passage

[1] We publish the reasons for removal, the removed, and remaining queries in the GitHub repository

sample improves the reliability of the labels, thus we aim for a trade-off between having multiple annotators per sample as well as annotating many samples in order to annotate with a high depth for each test query. In the literature there is a range from 1 annotator [Voo00] to multiple annotators per query-passage pair [HZS+20, CMYC19]. We aim for three relevance assessments per query-document pair, on average we reach 2.92 relevance assessments per pair, where we employ majority voting or a heuristic in case of no majority. We discard the pairs with only one relevance assessment (57 query-passage pairs in total). **3)** We monitor the average annotation time per pair per relevance grade, we remove annotations with a short annotation time (below 1 second) and reach an average annotation time of 48 seconds per judgement. The reason for choosing the threshold at 1 second is the distribution of annotation time, which follows a roughly normal distribution with mean at 48 seconds, but has a outlier at annotations with annotation time below 1 second. **4)** We collect feedback about the campaign from the students at the end of the annotation campaign. **5)** We conduct another annotation campaign with expert annotators and compare the relevance judgements of the experts with the non-expert annotators. **6)** We encourage the annotators to read up on concepts contained in the queries or documents that they are not familiar with using external tools like web search engines.

We conducted the annotation campaign using the FiRA interface [HZH20b, HZS+20] and ran the campaign for 7 days with a fixed deadline. We publish the annotation guidelines in our GitHub repository as well as in the Appendix of this thesis in Section 6.3. To control the quality of the judgements during the campaign, we monitored the average number of judgements per 12 hours and we observe the daily average annotation time per relevance grade to detect random judgements. We reach a high average annotation time per relevance grade (Table 5.1).

Overall the students gave us positive feedback about the evaluation campaign: 33% rated it *Very good*, 28% *Good* and 27% *Decent*, only 12% of the students did not like it. The students could also give written feedback. The main feedback was that it was hard to distinguish between the relevance grades of *Topic* and *Partial*. This difficulty is also reflected in the average time per annotation for these relevance grades. While it took the annotators on average 39 seconds to decide on *Wrong*, the annotations for *Topic* and *Partial* took 47 and 54 seconds on average, respectively. We reached 38810 total judgements and judged at least until the Top-4 for our pool.

5.1.4 Quality analysis

After the annotation campaign we processed the 38810 raw relevance judgements to form the TripJudge test collection which we publish in the standard TREC format for qrels.

We removed the query-document pairs with only one judgement. We grouped the relevance grades *Wrong* and *Topic* into *Irrelevant* and *Partial* and *Perfect* into *Relevant*, in order to attain a 2-graded relevance judgement with potential higher agreement. We computed the final relevance judgement either via full agreement, if all annotators agree on the relevance grade or with majority voting, if the annotators disagree, or with the heuristic of taking the lowest relevance grade, if the annotators disagree and there is no majority for a relevance grade. We apply this heuristic with the assumption that if disagreement is high, the relevance cannot be definitely decided and therefore the document should be annotated as irrelevant. In Figure 5.2 we visualize the 4-grade

# of queries	1136
# of documents in collection	2.3M
# annotated q-d pairs	12590
# of judgements	38810
avg # assessments per query	2.92
# (%), avg annotation time for Wrong	3811 (10%), 39s
# (%), avg annotation time for Topic	10901 (28%), 47s
# (%), avg annotation time for Partial	13008 (33%), 54s
# (%), avg annotation time for Perfect	11090 (29%), 44s
# annotators	135
avg # of annotations per annotator	287

Table 5.1: Statistics of the annotation campaign.

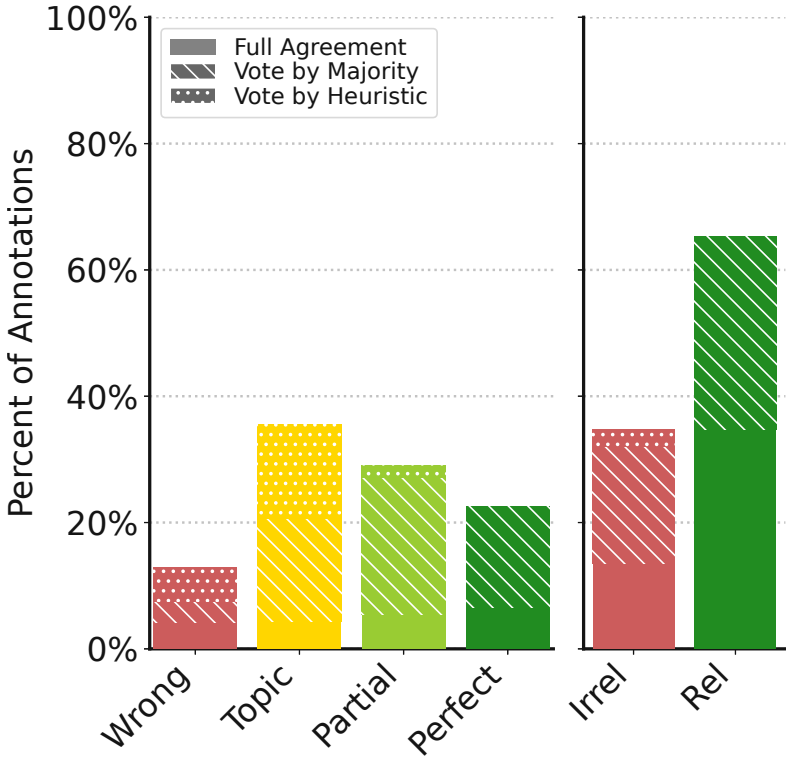


Figure 5.2: Distribution of relevance grades for 4-grade and 2-grades, percentage of heuristic and majority voting.

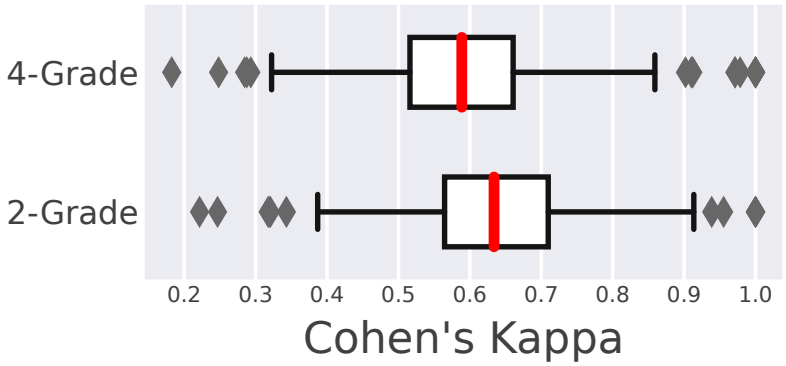


Figure 5.3: Cohen's Kappa agreement between the non-expert annotators and the annotations aggregated with majority voting.

and 2-grade distribution of the judgements and their percentage of full agreement, majority voting and the heuristic. While the full agreement is low (20% of all queries) for the 4-grade relevance judgements, the full agreement for the 2-grade relevance judgements is substantially higher with 48% of all queries. Furthermore, a high percentage of judgements is decided via majority vote. For the 4-grade relevance judgements, 22% of the queries are decided by the heuristic which indicates a high disagreement between the annotators, but the percentage of heuristic decisions for the 2-grade relevance judgements is low with 2%. This shows that the 2-grade relevance judgements are more robust and have a higher agreement between the annotators.

We also study the inter-annotator agreement between the annotators for all relevance judgements. We measure the inter-annotator agreement with Cohen's Kappa κ [Coh60], which is a standard metric to compare multiple sets of judgements and to measure the subjectivity in the assessments. As the 4-grade annotations are ordinal, we use a linear weighted Kappa for them.

In Figure 5.3 the 2-grade and 4-grade agreement with Cohen's Kappa is visualized with an average κ of 0.63 for the 2-grade and an average weighted κ of 0.60 for the 4-grade relevance judgements, which indicates moderate agreement for the 4-grade and substantial agreement for the 2-grade. For both the 4-grade and 2-grade agreement we reach an at least moderate agreement with 50% of the kappa values between 0.50 and 0.70. Furthermore, a certain level of disagreement in the relevance judgement is expected due to the subjectivity of the individual annotators [Bor03, VvdBWK17] and our agreement levels align with previous work [AM12], which also employs non-expert annotators for judgements of TREC collections.

5.1.5 Expert annotation campaign

In order to ensure and control the quality of non-expert annotations in TripJudge, we conduct another annotation campaign with experts from the medical and health domain. We compare the aggregated expert annotations to the non-expert annotations in TripJudge and measure the accuracy between the labels, similar to the approach of Snow et al. [SOJN08]. Snow et al. [SOJN08] evaluate non-expert annotations for natural language tasks including affect recognition,

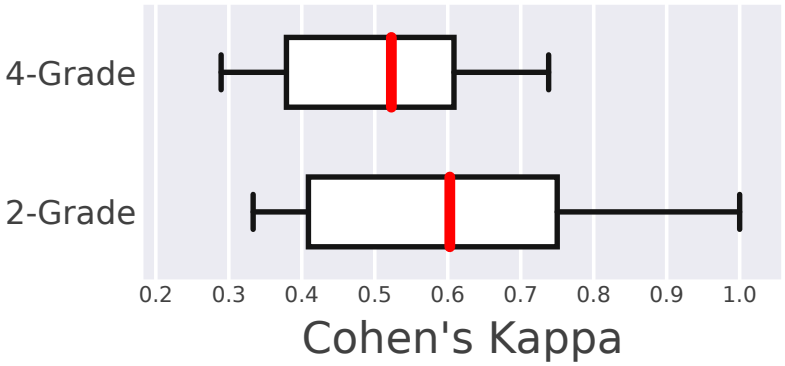


Figure 5.4: Cohen's Kappa agreement between the expert annotators and the annotations aggregated with majority voting.

word similarity, and recognizing textual entailment. They compare the non-expert annotations to gold standard labels by measuring the accuracy between both labels.

Since expert annotations are highly costly, we annotate a subset of the TripJudge test set, in order to evaluate, how much the non-expert annotations agree with the expert annotations. We conduct the expert annotation campaign using Prolific[2], which is a platform to recruit study participants for online research. In this platform it is possible to specify certain skill sets and employment characteristics of the study participants. In order to recruit study participants similar to the user distribution of the Trip search engine as well as to recruit participants that are experts in the health and medical domain, we specify certain characteristics that the study participants need to fulfill:

- The study participants need to have the current country of residence in either the United Kingdom or the United States of America,
- the study participants need to have English as their primary language,
- the study participants need to have education in health and medicine and
- the study participants need to be employed in the sector of medicine.

We have in total 12 study participants. For the annotation campaign we use the same annotation interface as for the campaign with the non-expert annotators with the same instructions.

Since expert annotations are highly costly, we annotate a randomly sampled subset of the TripJudge test collection. We randomly sample 50 query-passage pairs from the TripJudge test collection and collect 3 annotations per sample. We collect 150 relevance judgements in total and each annotator annotates maximally 15 query-passage pairs. Following the same methodology as in the non-expert annotation campaign, we aggregate the label from the expert annotations using the majority voting.

We also measure the inter-rater agreement of the expert annotators using Cohen's Kappa. In Figure 5.4 the 2-grade and 4-grade agreement of the expert annotations is visualized with Cohen's

[2] https://www.prolific.com/

Kappa. For the 2-graded relevance judgements we reach a median kappa score of 61.2% denoting substantial agreement of annotators. For the 4-graded relevance judgements we reach a moderate agreement of the annotators with a median kappa score of 52.2%. These agreement scores are relatively similar to the non-expert agreements and show that experts and non-experts similarly agree and disagree for this annotation task.

In order to compare the expert annotations with the non-expert annotations we compute the accuracy between the aggregated labels of the expert annotation campaign and the aggregated labels of TripJudge, following the approach of Snow et al. [SOJN08].

For the 2-graded relevance labels we reach an accuracy of 78% and for the 4-graded relevance labels we reach an accuracy of 48% between the aggregated expert and non-expert labels. This denotes that the experts and non-experts agree in 78% of the cases, deciding if a passage is relevant to a given query or not. This accuracy level aligns with similar accuracy values of Snow et al. [SOJN08], who measure an accuracy of 82% between non-expert and gold standard annotations for the task of recognizing textual entailment, when collecting 3 annotations per sample. It is to be expected that the accuracy for the 4-graded relevance is lower than the accuracy for the binary relevance judgements, since with 4 different classes the chances of disagreement are higher and only minor differences for example if the expert label is "Perfect" but the non-expert label is "Partial" are evaluated as disagreement. Since the experts and non-experts agree in the aggregated binary relevance judgements in more than 3 out of 4 cases, the non-expert annotations align in most cases with the expert annotations and can be used as labels to compare different ranking and retrieval systems. These results align with the results of Snow et al. [SOJN08], who also conclude that for many natural language annotation tasks a small number of non-expert annotations per sample is necessary to match the performance of an expert annotator.

5.1.6 TripJudge vs TripClick

We compare the relevance judgements of TripJudge with the click-based labels of TripClick and investigate the system ranking difference of the two test collections. Due to the higher inter-annotator agreement we consider the 2-graded judgements of TripJudge.

Coverage and Intersection

We analyze the coverage and intersection of the annotated query-document pairs between TripJudge and TripClick DCTR and Raw test collection. Figure 5.5 visualizes the percentage of relevant and irrelevant relevance judgements from TripJudge. The different patterns of the bar visualize the label from the TripClick DCTR or Raw test collection. The labels 1/2/3 from TripClick refer to relevant documents, Label 0 denotes irrelevant documents and unlabelled documents are considered as irrelevant during evaluation. The green bars denote agreement between the annotation from TripJudge and TripClick, the red bars denote disagreement. It is striking that all of the relevant documents from the Top-4 of TripJudge are unlabelled in DCTR and therefore considered as irrelevant. Furthermore there is high disagreement between the judgements of TripJudge and click-based labels of TripClick DCTR and Raw.

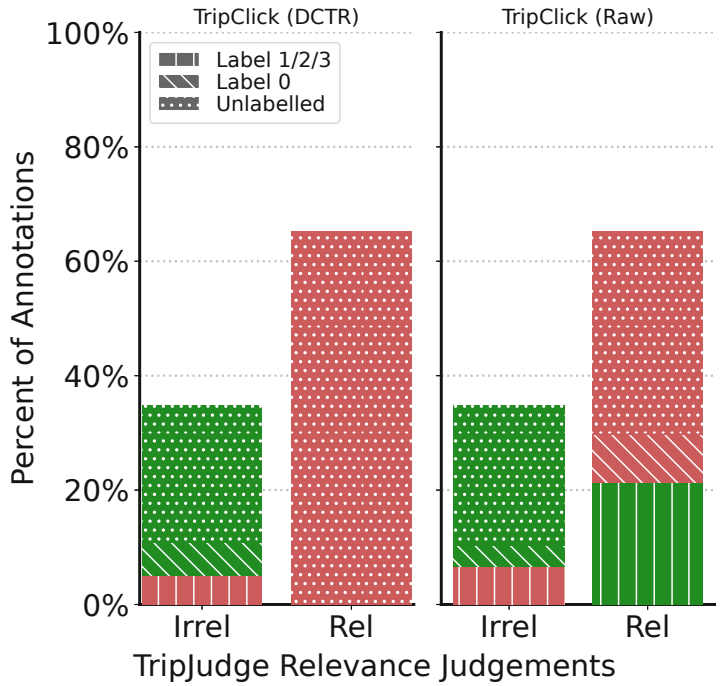


Figure 5.5: Relevance judgements from TripJudge for the Top-4 of the pool vs TripClick click-based labels from the DCTR or Raw test collection. Green bars denote agreement between the relevance judgement of TripJudge and the click label from TripClick, red bars denote disagreement.

Model	TripJudge						TripClick (DCTR)						TripClick (Raw)					
	J@5	J@10	n@5-j	n@5	n@10	R@100	J@5	J@10	n@5	n@10	R@100		J@5	J@10	n@5	n@10	R@100	
First stage retrieval																		
1 BM25	78%	47%	**.761**	.694	.570	.771	33%	31%	.122	.140	.499		30%	27%	.199	.198	.464	
2 DPR SciBERT	87%	50%	.602	.540	.456	.636	48%	41%	.232	.243	.562		44%	38%	.362	.328	.496	
3 DPR PMBERT	49%	36%	.652	.377	.356	.649	45%	40%	.223	.235	.582		42%	37%	.345	.318	.518	
Re-Ranking (BM25 Top-200)																		
4 ColSciBERT	64%	44%	.758	.538	.501	.790	52%	47%	.254	.270	.589		49%	43%	.395	.367	.547	
5 ColPMBERT	63%	44%	.758	.527	.493	.777	55%	49%	.261	.278	.595		52%	45%	.412	.382	.551	
6 MonoBERT	64%	45%	.757	.540	.506	**.818**	56%	50%	.271	.287	.594		53%	46%	.421	.389	.552	
7 Ensemble	**88%**	**51%**	.756	**.698**	**.592**	.814	**58%**	**52%**	**.285**	**.303**	**.600**		**55%**	**48%**	**.443**	**.409**	**.556**	

Table 5.2: Effectiveness results and judgement coverage for judgement-based TripJudge and click-based TripClick DCTR/Raw test collection. J@m denotes the judgement coverage at rank m, n@m denotes the nDCG at cutoff m, -j denotes the j-option in trec_eval when only evaluating on the judged query-document pairs. Top-10 of run 1,2,7 create the pool for TripJudge.

Measure	TripJudge–DCTR	TripJudge–Raw	DCTR–Raw
nDCG@5	0.333	0.333	1.000
nDCG@10	0.428	0.428	1.000
MRR@10	0.238	0.238	1.000
Recall@100	0.238	0.333	0.714

Table 5.3: Kendall tau correlation between system rankings of TripJudge and TripClick DCTR/Raw for four metrics.

Qualitative analysis of the discrepancies

In order to investigate potential reasons for the discrepancies between the click-based and human-annotated labels, we conduct a qualitative analysis. For that we look at 5 randomly sampled query-document pairs, that are annotated as relevant in TripJudge, but are labelled as irrelevant by the click-labels from TripClick (DCTR) and TripClick (Raw). In the following list are the 5 randomly sampled query-document pairs with the query IDs, query text, document IDs and the document text, which is the beginning snippet of the document text.

Example 1

Query ID: 13739

Query: gestational diabetes mellitus

Document ID: 8991404

Document text: The Comparative Effectiveness of Diabetes Prevention Strategies to Reduce Postpartum Weight Retention in Women With Gestational Diabetes Mellitus: The Gestational Diabetes' Effects on Moms (GEM) Cluster Randomized Controlled Trial OBJECTIVE : To compare the effectiveness of diabetes prevention strategies addressing postpartum weight retention for women with gestational diabetes mellitus (GDM) delivered at the health system level: mailed recommendations (usual care) versus usual care plus a Diabetes Prevention Program (DPP)-derived lifestyle intervention. RESEARCH DESIGN AND METHODS : This study was a cluster randomized controlled trial of 44 medical facilities (including 2,280 women with GDM) randomized to intervention or usual care. The intervention included mailed gestational weight gain recommendations plus 13 telephone sessions between 6 weeks and 6 months postpartum...

Example 2

Query ID: 1291

Query: macular degeneration

Document ID: 9336985

Document text: Statins for age-related macular degeneration. BACKGROUND : Age-related macular degeneration (AMD) is a progressive, late-onset disorder of the macula affecting central vision. It is the leading cause of blindness in people over 65 years in industrialized countries. Recent epidemiologic, genetic, and pathological evidence has shown that AMD shares a number of risk factors with atherosclerosis, leading to the hypothesis that statins may exert protective effects in AMD. OBJECTIVES : The objective of this review was to examine the effectiveness of statins compared with other treatments, no treatment, or placebo in delaying the onset and progression of AMD. SEARCH METHODS : We searched the Cochrane Central Register of Controlled Trials...

Example 3

Query ID: 4284

Query: postnatal depression

Document ID: 9089460

Document text: Identification of depression in women during pregnancy and the early postnatal period using the Whooley questions and the Edinburgh Postnatal Depression Scale: protocol for the Born and Bred in Yorkshire: PeriNatal Depression Diagnostic Accuracy (BaBY PaN INTRODUCTION : Perinatal depression is well recognised as a mental health condition but <50% of cases are identified by healthcare professionals in routine clinical practice. The Edinburgh Postnatal Depression Scale (EPDS) is often used to detect symptoms of postnatal depression in maternity and child services. The National Institute for Health and Care Excellence (NICE) recommends 2 'ultra-brief' case-finding questions (the Whooley questions) to aid identification of depression during the perinatal period, but this recommendation was made in the absence of any validation studies in a perinatal population...

Example 4

Query ID: 1585386

Query: rheumatoid arthritis juvenile arthritis juvenile idiopathic arthritis air pollut

Document ID: 9569762

Document text: Ambient air pollution exposures and risk of rheumatoid arthritis. OBJECTIVE : Environmental factors may play a role in the development of rheumatoid arthritis (RA). We previously observed increased RA risk among women living closer to major roads (a source of air pollution). Herein, we examined whether long-term exposures to specific air pollutants were associated with RA risk among women in the Nurses' Health Study (NHS). METHODS : The NHS is a large US cohort of female nurses followed up prospectively every 2 years since 1976. We studied 111,425 NHS participants with information on air pollution exposures as well as data concerning other lifestyle and behavioral exposures and disease outcomes. Outdoor levels of different size fractions of particulate matter (PM10 and

PM2.5) and gaseous pollutants (SO2 and NO2) were predicted for all available residential addresses using monitoring data from the US Environmental Protection Agency. We examined the association of time-varying exposures 6 and 10 years before each questionnaire cycle and cumulative average exposure with the risk of RA, seronegative (rheumatoid factor and anti-citrullinated peptide antibody negative) RA, and seropositive RA...

Example 5

Query ID: 47480

Query: nausea vomiting pregnancy

Document ID: 7779586

Document text: Pregnancy complications and birth outcomes among women experiencing nausea only or nausea and vomiting during pregnancy in the Norwegian Mother and Child Cohort Study BACKGROUND : To compare pregnancy complications and birth outcomes for women experiencing nausea and vomiting in pregnancy, or nausea only, with symptom-free women. METHODS : Pregnancies from the Norwegian Mother and Child Cohort Study (n = 51 675), a population-based prospective cohort study, were examined. Data on nausea and/or vomiting during gestation and birth outcomes were collected from three questionnaires answered between gestation weeks 15 and 30, and linked with data from the Medical Birth Registry of Norway. Chi-squared tests, one way analysis of variance, multiple linear and logistic regression analyses were used. RESULTS : Women with nausea and vomiting (NVP) totalled 17 070 (33%), while 20 371 (39%) experienced nausea only (NP), and 14 234 (28%) were symptom-free (SF)...

When analyzing **Example 1** we see that the user query is "gestational diabetes mellitus" and the document, that is annotated as relevant in TripJudge, is a study about different prevention strategies to reduce postpartum weight retention in women with gestational diabetes mellitus. While the document includes patients with gestational diabetes mellitus, it is only related to the topic and does not mainly talk about gestational diabetes mellitus.

In **Example 2** the user query is "macular degeneration" and the document, that is annotated as relevant in TripJudge, is a study about statins for age-related macular degeneration, thus we would annotate that document as relevant to the query although it was not clicked by users of Trip.

In **Example 3** the user query is "postnatal depression" and the document is a study about identifying depression in women during pregnancy and the early postnatal period, thus the document is related to postnatal depression and is relevant for the query, different to the label of the TripClick test set.

In **Example 4** the query is a combination of the key word "rheumatoid arthritis juvenile idiopathic arthritis air pollut". The user probably wants to find studies investigating relations of rheumatoid arthritis or juvenile idiopathic arthritis with air pollution. Thus the study about air pollution

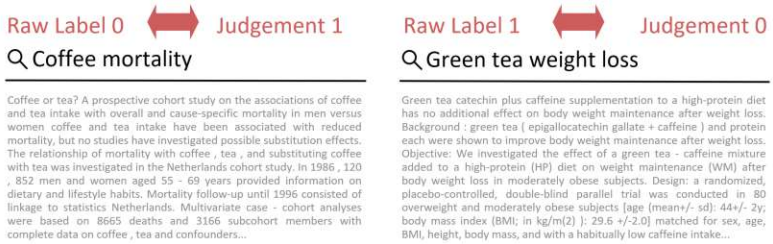


Figure 5.6: Two examples of query-document pairs where the TripClick (Raw) label and the TripJudge label disagree. ON the left side the click-label is 0 (irrelevant), but the TripJudge judgement is 1 (relevant), on the right side it is the opposite case: The TripClick label is 1 (relevant), but the TripJudge label is 0 (irrelevant).

exposures and risks of rheumatoid arthritis seems like a relevant document, although it was not clicked by the users of Trip.

In **Example 5** the user query is "nausea vomitting pregnancy" and the document is about pregnancy complications and birth outcomes of women who experience nausea or vomiting during pregnancy. While this study might not be the most relevant document for this search query, we would still say it is a relevant study for this query.

Furthermore we visualize another disagreement example in Figure 5.6. On the left side of the Figure 5.6 we see an example of a query and a document, where the raw label of the TripClick (Raw) dataset is 0 (irrelevant) but the judgement in TripJudge for this query-document pair is 1 (relevant). On the right side of the Figure, we see the opposite case: The TripClick (Raw) label is 1 (relevant), but the judgement annotation in TripJudge is 0 (irrelevant). On the left side the user query is "coffee mortality" and the document looks like a study investigating the effect of coffee and tea intake on the overall and cause-specific mortality of men and women. Thus this seems to be a study drawings connections between coffee and mortality and the author would also denote this study as relevant, although it was not clicked by the users. On the right side the user query is "green tea weight loss", where the users intent is probably to find studies that investigate the connection of green tea and weight loss. However the document is a study about the effect of green tea plus coffee intake on body weight maintenance. It also takes into account that the participants first have lost weight, thus the document text contains the words "weight loss", but the study does not investigate direct connections of green tea and weight loss, but connections of green tea and weight maintenance. Thus we would also not consider this document as relevant.

This qualitative analysis reveals that for 4 out of 5 random examples the judgements of TripJudge make sense and the author agrees with the label of TripJudge and disagrees with the click-label from TripClick. Furthermore we analyze two more examples where the click-based and human-annotation-based label disagree and find that the human-label seems to be more in line with the judgement of relevance of the author. That sheds a light on the discrepancies between the labels of TripJudge and TripClick that are visualized in Figure 5.5 and shows that a non-click does not automatically mean that the document is not relevant to the query.

System evaluation

We compare the system rankings and the coverage of the relevance judgements for the runs for TripJudge and TripClick DCTR and Raw. For TripJudge and TripClick unlabelled documents are considered as irrelevant. In Table 5.2 the effectiveness metrics as well as the judgements coverage measured as J at rank n is displayed for various statistical and neural retrieval systems from Hofstätter et al. [HASH22]. For TripJudge we see that the coverage measure with J@5 for the runs in the pool (run 1,2,7) is high (around 80%) compared to the coverage of the runs which did not participate in the pooling. However the coverage of TripJudge is higher than the coverage of TripClick DCTR and Raw for the respective runs. The Ensemble of MonoBERT based on SciBERT, PubMedBERT-Abstract and PubMedBert-Full Text (run 7) consistently reaches the highest retrieval performance for the judgement and click-based test collections. Interestingly the dense retrieval model DPR SciBERT (run 2) underperforms BM25 (run 1) when evaluated with TripJudge, although showing substantially higher retrieval effectiveness for the click-based test collections. This suggests that the higher retrieval effectiveness of run 2 compared to run 1 for the click-based collections is due to the higher coverage of annotations.

Furthermore, we compare the difference in system rankings between two test collection with Kendall τ [Ken38], which is a common measure to compare the correlation between two system rankings [Voo98, SS07]. In Table 5.3 are the Kendall tau correlations between two rankings of two test collections regarding four metrics. Test collections with $\tau > 0.9$ are considered equivalent [Voo98].

For the comparison of TripJudge with the click-based test collections, we see a low correlation of the system rankings for all 4 evaluation metrics. For the click-based test collections we see that they are equivalent for most of the metrics. We conclude that the system rankings differ drastically between TripJudge and TripClick and that TripJudge offers a valuable and reusable relevance judgement set beside the TripClick test sets for evaluating retrieval systems.

5.1.7 Conclusion

We present the TripJudge test collection with 38810 relevance judgements for TripClick health retrieval. We describe the annotation campaign for creating the relevance judgements. For increased re-usability we used lexical and neural retrieval systems for pool creation. We reach an at least moderate inter-annotator agreement among non-expert annotators. In order to control the quality of the non-expert annotations in TripJudge, we compare the non-expert annotations with expert annotations on a subset of the TripJudge test set and find a high overlap between the relevance labels. When comparing the relevance judgements of TripJudge with the click-based annotations from the TripClick test collections, we find that a majority of judged relevant documents were unlabelled in TripClick, and there is a high disagreement between the relevance judgements and the click-based annotations. We re-evaluate lexical and neural models and find a higher judgement coverage for the retrieval runs for TripJudge than for the TripClick test collections. The system rankings substantially differ between the evaluation with the relevance-based and click-based collections.

Limitations and future work

One limitation of TripJudge is the depth of the relevance judgements is usually a half [CMY+21b] or a third of the annotated query-document pairs should be relevant [Voo18, VR21] while we have 65% relevant. Therefore we see possible future work in annotating to a higher depth. Nevertheless we view TripJudge as a valuable resource for evaluation of the domain specific task of health retrieval. Another limitations is that TripJudge is annotated by non-experts and the whole test set is not annotated by experts. To mitigate this limitation and in order to control the quality of the non-expert annotations we conduct another expert annotation campaign and find a high overlap of expert and non-expert annotations. Still it is a limitation of that study that we mainly employed non-expert annotators. Another limitation is the number of annotators per sample. We had to manage a trade-off between annotating at least all samples from the test set with the number of annotators and thus decided on a compromise of 3 annotators per query-document sample, while of course more annotators per query-document sample would improve the quality of the test set. Furthermore one direction that was not investigated in this work is the choice of aggregation method for the annotations per query-document sample. While majority voting is a rather simplistic choice there are more elaborate heuristics for aggregating the labels, that can be studied in future work.

In conclusion, we argue that there must be more effort put into creating relevance judgements based test collections for domain specific retrieval tasks, in order to evaluate different systems in a robust and conclusive way.

5.2 Active Learning for Annotation Efficiency Improvements

Having investigated the difference between human-labelled and click-based relevance annotations for evaluation of medical ad-hoc retrieval, we now want to focus on exploring the problem of training neural rankers with limited training data. We investigate:

RQ2.2 To what extent does active learning improve annotation efficiency for training neural ranking and retrieval models?

We study this research question in the context of the tasks of ad-hoc retrieval in the web and medical domain. In our research, we aim to isolate the impact of active learning for efficient training data annotation, distinct from the potential influence of domain-specific neural ranking and retrieval architectures like BERT-PLI [SML+20] or PARM. To achieve this, we focus on "common" neural ranking and retrieval models [NC19, KOM+20, KZ20], which are not explicitly designed for long documents. This chapter centers on ad-hoc retrieval tasks in the web and medical domains, where long documents are not part of the queries or collections, obviating the need for architectural modifications in the models introduced in Chapter 2.6. Also we choose to focus on medical and web ad-hoc retrieval tasks because of the accessibility of large-scale, annotated training data for these tasks, including MS Marco [NRS+16] and TripClick [RLS+21]. Due to resource limitations, we simulate the selection and annotation processes during training and thus need large-scale already annotated training datasets for this simulation. Our evaluation focuses on precision-oriented metrics, given the precision-focused nature of ad-hoc retrieval tasks. This chapter is based on the publication [AZH+23].

Search methods based on Pre-trained Language Models (PLM) have demonstrated great effectiveness gains compared to statistical and early neural ranking models [CMYC19, CMYC20]. However, fine-tuning neural rankers requires a great amount of annotated training data [NRS+16, CMYC20]. Annotating data involves a large manual effort and thus is expensive, especially in domain specific tasks [CBLL20]. In this chapter we consider the problem of fine-tuning neural rankers under limited training data and budget. We investigate two scenarios: fine-tuning a ranker from scratch, and domain adaptation starting with a ranker already fine-tuned on general data, and continuing fine-tuning on a target dataset.

We observe a great variability in effectiveness when fine-tuning on different randomly selected subsets of training data. This suggests that it is possible to achieve effectiveness gains by actively selecting a subset of the training data that has the most positive effect on the rankers. This way, it would be possible to fine-tune effective neural rankers at a reduced annotation budget. To investigate this, we adapt existing Active Learning (AL) strategies to the task of fine-tuning neural rankers and investigate their effectiveness, also considering annotation and computational costs. Our extensive analysis shows that AL strategies do not significantly outperform random selection of training subsets in terms of effectiveness. We further find that gains provided by AL strategies come at the expense of more assessments (thus higher annotation costs) and AL strategies underperform random selection when comparing effectiveness given a fixed annotation cost. Our results highlight that “optimal” subsets of training data that provide high effectiveness

at low annotation cost do exist, but current mainstream AL strategies applied to neural rankers are not capable of identifying them.

5.2.1 Introduction

Search methods based on large Pre-trained Language Models (PLM) have shown great effectiveness gains compared to common statistical models and early neural methods [LNY21, CMY+21b, Ton22]. These language models are pre-trained for language representation learning on a background corpus; they are then further trained for a specific task – a process referred to as fine-tuning. Typically, neural rankers are created through the fine-tuning of a PLM to the ranking task (and possibly, to a specific domain) [NC19, NYCL19]. The fine-tuning of neural rankers typically requires a great amount of labelled training data [NC19, HLY+21]. This can be a challenge when considering search tasks with no or little training data available. Data annotation typically requires a large manual effort and thus is expensive, especially in domain-specific tasks where annotators should be domain experts [CBLL20]. In real-life settings, annotation and computational budget[3] is often limited, especially for start-ups or in domain-specific contexts [Tai14].

In this work we focus on the problem of fine-tuning neural rankers under limited training data and budget. There are alternative directions one may take to deploy a neural ranker in a specific task for which no or limited training data is available. These include for example the zero-shot application of neural rankers trained on another, resource-rich, retrieval task or domain [XXS+22, TRR+21], the learning with few-shot examples [DZM+23], and approaches based on pseudo-labelling [WTRG22]. However the effectiveness of these approaches depends on the relatedness of the fine-tuning task or the pre-training domain of the language model to the target retrieval task [WSKZ22]; thus their generalization capabilities remain unclear. Therefore performing domain adaptation by fine-tuning the neural ranker on the target task with annotated training data (the setting investigated in this work) remains favourable for a (reliable) high effectiveness [CMY+21a].

It is unclear however how much annotated training data is required for training an effective neural ranker. Furthermore, in presence of a budget constraint that restricts the amount of data that can be annotated for training, it is unclear whether it is possible to select training data to minimise annotation cost while maximising ranker effectiveness.

In this work, (1) we investigate how the amount of labelled data used for fine-tuning a neural ranker impacts its effectiveness, (2) we adapt active learning (AL) strategies to the task of training neural rankers, (3) we propose a budget-aware evaluation schema including aspects of annotation and computation cost, (4) we conduct an extensive analysis of AL strategies for training neural rankers investigating the trade-offs between effectiveness, annotation budget and computational budget. We do this in the context of three common neural ranker architectures: cross-encoders (MonoBERT [NC19]), single representation bi-encoders (DPR [KOM+20]) and multi-representation bi-encoders (ColBERT [KZ20]), and two scenarios:

[3] With annotation budget we refer to the amount of money set aside for paying annotators to label pairs of queries and documents. With computational budget, we refer to the amount of money set aside for paying the computation costs arising from the training/fine-tuning of the neural rankers. These costs may include the hardware and energy costs, or the purchase of cloud solutions.

❶ **Scratch**: the PLM is pre-trained on a background corpus, but has yet to be fine-tuned to the target ranking task and dataset;

❷ **Adapt**: domain adaptation of the neural ranker is performed. The PLM is pre-trained on a background corpus and fine-tuned to a ranking task and a specific dataset, but further fine-tuning has yet to be performed to transfer the ranker to another dataset and, possibly, a ranking task with characteristics that differ from those of the first fine-tuning process.

In our study we investigate the following research question:

RQ2.2.1 What is the effect of the size of the labelled training data on the effectiveness of neural rankers?

To investigate the effect of the amount of labelled data on the effectiveness of neural rankers, we select incremental amounts of data to fine-tune a ranker. Our empirical results show that the size of the dataset available for fine-tuning the neural ranker greatly influences the effectiveness of the ranker. While, somewhat unsurprisingly, we find that in general more training data leads to higher effectiveness, we also find large variability in effectiveness between different randomly selected training sets of the same size. Furthermore we find that, for some training sizes, the best random selection run outperforms the worst one, and significantly. This shows that there are subsets of the training data which lead to significant improvements within the same training data size.

This variability motivates us to investigate whether we can select those "high-yield" samples using Active Learning (AL) strategies. The intuition is that a good selection strategy would lead to a smaller amount of data to be annotated, and thus a lower annotation cost, while still producing a highly effective ranker. We investigate:

RQ2.2.2 How do different active selection strategies influence the effectiveness of neural rankers?

Selection of training data has been extensively investigated in AL for machine learning. Here, common active selection strategies are based on uncertainty or diversity criteria [CGJ96, LG94, SZ05]. We thus adapt representative methods that implement these criteria to the context of fine-tuning neural rankers. We evaluate the representative active selection strategies in terms of their effectiveness for fine-tuning neural rankers on different training data sizes and compare the strategies to random selection of training data as baseline. For both scenarios the active selection strategies do not offer statistically significant improvements compared to random selection. For certain scenarios and neural rankers we find varying beneficial selection strategies, however no selection strategy shows consistent and robust higher effectiveness than random selection.

Since it is not our goal to minimize the training data size, but actually we aim to minimize the total cost of fine-tuning neural rankers, we investigate:

RQ2.2.3 What is the effect of using an an active selection strategy to fine-tune a neural ranker under a constrained budget?

We revisit the results in light of a budget-aware evaluation which we introduce in this work. This evaluation includes aspects of annotation cost as well as cost of computing resources. With this,

we find that the annotations are the main cost factor. Since each selection methods requires a different number of assessments to annotate a training set of a certain size, we compare the number of assessments to the effectiveness of the neural rankers for random and active selection strategies. This reveals that the (marginal, if any) effectiveness gains provided by AL strategies come at the expense of more assessments (thus higher annotation costs) and AL strategies under-perform random selection when comparing both effectiveness and associated cost.

Our contributions are the following:

- We find a great variability in effectiveness when fine-tuning a neural ranker on different subsets of the same size;
- We adapt active selection strategies to the task of training neural rankers;
- We propose our novel budget-aware evaluation schema including aspects of annotation cost and cost for computational resources;
- We conduct an extensive analysis of active learning strategies for training neural rankers for two different training scenarios in the context of budget-aware evaluation.

5.2.2 Considered neural rankers

In this study we consider three neural ranker architectures considered: a cross-encoder model (MonoBERT) [NC19], a single representation bi-encoder model (DPR) [KOM+20], and a multi representation bi-encoder model (ColBERT) [KZ20]. We refer to details about the model architectures to our background section 2.

5.2.3 Training Scenarios & Annotation Modeling

We consider two scenarios for training the neural rankers: ❶ **Scratch** training from scratch, starting with a PLM and ❷ **Adapt** domain fine-tuning after rank/retrieval fine-tuning of the PLM. These are common scenarios that are encountered in the practical application of neural rankers to search problems [DZM+23, HLY+21, HASH22].

In ❶ **Scratch** our objective is to train a neural ranker "from scratch", i.e., without having already performed any fine-tuning on a retrieval task. There are many reasons this scenario could occur in the practical deployment of neural rankers. For example, no suitable labelled data corresponding to the ranking task may be available, or the data that may be available is protected by a license that prevents its use within a product (e.g., the MS Marco dataset). We model the first scenario by starting from a pre-trained BERT model [DCLT19, SDCW19] and conduct experiments by training the ranker on the MS Marco dataset, a large scale web search collection commonly used to train these rankers [NC19, HLY+21, QDL+21]. Note, in our experiments we assume that no labels are available for the dataset, and labels are iteratively collected (in a simulated setting) within the AL cycle.

In ❷ **Adapt** our goal is to adapt a neural ranker to a specific retrieval task (potentially in a specific domain). Here we assume that the neural ranker has already undergone fine-tuning on

Algorithm 2 Incremental annotation and training process

Input: T whole training set, I number of iterations, s number of added samples per iteration, M is a neural ranker/retriever
Output: D annotated training set, M neural ranker trained on D

```
D ← {}
for i in I do
    Select subset S ⊂ T of size |S| = s with selection strategy
    Annotate S
    D ← D ∪ S
    Train M with D
end
```

a high-resource retrieval task (e.g., using the common MS Marco dataset [NRS+16, KOM+20, HLY+21, QDL+21]), and the goal is to further fine-tune the ranker with additional data, on a different retrieval task or data domain. This is a common setting in domain-specific IR settings [DZM+23, WTRG22, HASH22]. The assumption is that the initial fine-tuning on the non-target retrieval task or domain data still highly contributes to the effectiveness of the ranker, especially when the target data available for fine-tuning is limited. We model the second scenario by starting from a ranking or retrieval model fine-tuned on MS Marco and fine-tune the model for a domain-specific retrieval task. In our experiments, we choose to validate the models using the retrieval task and datasets associated with health-oriented web search in the medical domain [GJK+16]. We choose this task due to the availability of the TripClick dataset [RLS+21], a large-scale training and test set for this task, and the availability of the TripJudge test collection [AHVH22]. The TripClick dataset has similar characteristics to MS Marco (e.g. query length, sparse judgments). In contrast to other domain adaptation approaches [CGZW11], we do not mix the training sets of the source and the target domain. Instead, in our experiments, we only rely on the availability of the neural ranker already fine tuned on the non-target data. We do this so to (i) be able to separate the effects of mixing the training sets from the active domain adaptation strategies, and (ii) study neural ranker development and deployment strategies that are in line with the green IR principles of reuse and recycle [SZZ22].

In order to model the real-life process of incremental annotation and training we incrementally increase our training set D used for fine-tuning. The details of this incremental process are depicted in Algorithm 2. We start with an empty training set $D = \{\}$ and in each iteration a subset S of the whole training set T ($S \subset T$) is selected to be added to the training dataset D. We model the annotation process by attaining the labels from the training set qrels and adding the samples to the training set ($D = D \cup S$). Then we train the neural ranker on the updated training set D and, based on random or active selection strategies, we select the next subset to annotate and add it to the training set.

5.2.4 Active Selection Strategies

We consider three active selection strategies to identify training data for labelling and to then use within the neural ranker fine-tuning: uncertainty-based selection [LG94, ZWYT08, Yu05], query-by-committee (QBC) [CGZW11], and diversity-based selection [XAZ07]. In addition, we consider random selection as a baseline selection strategy. Next, we describe the active selection strategies and how we adapt them for fine-tuning neural rankers.

Uncertainty-based selection

The uncertainty-based selection strategy selects samples by measuring the model's (ranker) uncertainty in the scores it produced and then selecting the samples with the least confidence [LG94, CGJ96]. Uncertainty-based strategies are commonly applied to classification problems, and often the score provided by the classifier is used as direct indication of uncertainty [EDHG+20, LG94]: scores are in the range $[0, 1]$, the decision boundary is set to 0.5 and the confidence in the classification is measured in function of the distance to the decision boundary (the closer, the least confident) [EDHG+20, LG94].

This approach is not directly transferable to neural rankers since their relevance scores are not necessarily bounded and therefore there is no clear decision boundary measuring the uncertainty in the ranking. We note that uncertainty estimation in Information Retrieval is a fundamental but largely unexplored problem [TC96, CLvR98, CC07], especially for neural rankers [LRC+21, CML+21].

In this work, we model the decision boundary by the mean of the score distribution of the top K ranked passages for all queries in the training set $T \backslash D$ and select the query-passage pairs with the relevance score closest to the mean of the score distribution for the ranker, hence with the highest uncertainty. We leave the investigation of other upcoming approaches for future work (see Section 5.2.8 for further insights).

In order to model the annotation and training process for the selected query-passage pairs, the selected passage is assigned its label from the training qrels. In case the selected passage is *relevant* we sample an irrelevant passage randomly from the BM25 top 1,000 to construct a training triplet (query, positive passage, negative passage). In case the selected passage is *irrelevant*, we take the selected passage as negative for the triplet and take the first relevant passage in the BM25 top100 list re-ranked by the neural ranker as positive passage. In case the positive passage is not in the BM25 top 100 we still add the positive passage from the training qrels. For each selected query-passage pair we add one triplet to the train set.

Query-by-committee selection

The query-by-committee (QBC) method [SOS92, FSST97] is a specific uncertainty-based selection strategy. In QBC, multiple committee members (models) are used to classify or rank samples; then the disagreement between the committee members on classified/ranked samples is measured and the samples with the highest disagreement are selected for annotation. A previous adaptation of QBC to information retrieval is due to Cai et al. [CGZW11] who apply QBC to

Learning-to-Rank for domain adaptation. For this, they train different members of the committee by training on subsets of the currently annotated training set at hand. The disagreement between the committee members is then measured by the vote entropy of the different rankings of the members for the queries which are not yet annotated. In the vote entropy the committee members M vote on the partial order of two passages $N(p_1 \prec p_2)$ in the ranked list R, counting how many of the members rank p_1 higher than p_2. The vote entropy of a query q is then defined as:

$$VE(q) = \frac{-1}{|M|} \sum_{p_i, p_j \in R} N(p_i \prec p_j) log \left(\frac{N(p_i \prec p_j)}{|M|} \right) \tag{5.1}$$

The queries with the highest vote entropy are selected for annotation. In order to model the annotation process and to select training triples, for every query that is selected to add to the training set we take the first relevant passage in the BM25 top100 list re-ranked by the neural ranker as positive passage and sample a random negative passage from the BM25 top 1,000 passages. We choose to use the re-ranked list of the first member for selection. For every query we add one training triplet to the training set.

Diversity-based selection

Diversity-based selection strategies select training samples based on the diversity of the samples – typically comparing already selected samples to those yet to select [XAZ07]. Within Information Retrieval, diversity-based selection strategies have been used for Learning-to-Rank models [XAZ07, SZ05, YWGH09]. In those settings, diversity is measured by clustering queries using an external unsupervised clustering model and taking one representative query from each cluster.

In our adaptation of diversity-based selection strategies to neural rankers, to compute diversity we consider the query representation made by the neural ranker. This has the advantage that we leverage the model's representations to compute diversity, instead of relying on an external model. Furthermore, this representation changes in each iteration as the neural ranker is trained incrementally through training sets of increasing size: therefore, the query representation also accounts for changes within the ranker itself. For DPR and ColBERT, we use the CLS token representation of the encoded query as query representation. For MonoBERT we encode the query without a passage and also take the CLS representation for measuring the diversity. We cluster the query representations of queries in $T \backslash D$ with the number of clusters equaling the number of training samples to be added in that iteration (s). From each cluster, we randomly sample one query to be annotated.

To add a training triplet to the training set for each selected query, we rely on the same annotation process used in QBC.

5.2.5 Budget-aware evaluation

Next we introduce a framework to evaluate active learning to neural ranker fine-tuning within budget constraints. For this, we model the costs related to both annotation effort and computation.

Annotation Costs

For measuring the annotation costs, we count the number of assessments needed to annotate a training triplet for a single query. The number of assessments corresponds to the rank of the first relevant passage found in the ranking of the neural ranker, which is trained in the previous iteration of the AL process. In our setting, the annotation for a query stops when one relevant passage for a selected query is found – thus we need to assess all passages in the ranking for that query up until the relevant passage is found and also annotated. This implies that the number of assessments differs from the training data size (the number of queries), e.g., one query sample in the training data can account for 10 assessments required when the first relevant passage is found at rank 10.

Then total annotation cost is the sum of the number of assessments for all the queries added to the training set multiplied with the costs for annotating them. Formally, let $A(i)$ be the number of assessments needed to create the training data of iteration i, A_h be the number of assessments an annotator finishes in one hour and A_C be the cost for an annotator per hour[4]. Then, the total annotation cost at iteration i is computed as:

$$C_A(i) = \frac{A(i)}{A_h} \cdot A_C \tag{5.2}$$

Computational Costs

Next we model the computational costs involved in executing the active selection strategies. These strategies usually will require both CPU and GPU based computation, which incur different costs and thus we account for separately. Let $H_{GPU}(i)$ be the accumulated number of GPU hours needed for training a neural ranker for iteration i and G_h be the cost of running an GPU for one hour. Then, the total computational cost at iteration i is computed as:

$$C_C(i) = H_{GPU}(i) \cdot G_h + H_{CPU} \cdot C_h \cdot (i-1) \tag{5.3}$$

with H_{CPU} the number of CPU hours needed for computing the selection strategy and C_h the cost of one hour CPU.

Total Cost

Finally, the total cost at iteration i can then be computed using Equations 5.2 and 5.3:

$$\begin{aligned} C(i) &= C_A(i) + C_C(i) \\ &= \frac{A(i)}{A_h} \cdot A_C + H_{GPU}(i) \cdot G_h + H_{CPU} \cdot C_h \cdot (i-1) \end{aligned} \tag{5.4}$$

[4]Note that certain search tasks or domain may require multiple annotators to examine the same sample: in this case A_C would be the sum of the hourly rates associated to all the annotators.

5.2.6 Experimental Setup

Next we describe the experimental setup we have devised to study the AL strategies for neural ranker fine-tuning we have illustrated above. We develop our investigation along the following three lines of inquiry:

RQ2.2.1 What is the effect of the size of the labelled training data on the effectiveness of neural rankers?

RQ2.2.2 How do different active selection strategies influence the effectiveness of neural rankers?

RQ2.2.3 What is the effect of using an an active selection strategy to fine-tune a neural ranker under a constrained budget?

Passage Collection & Query Sets

For ❶ **Scratch**, we use the MS Marco passage collection [NRS+16] as described in section 2.5. MS Marco is based on sampled Bing queries and contains 8.8 million passages; its training set contains $503k$ training triplets. We use the training portion for fine-tuning and evaluate on the TREC DL 2019 [CMYC19] and 2020 [CMYC20] with nDCG@10.

For ❷ **Adapt**, we use the TripClick dataset [RLS+21] as described in section 2.5 and our TripJudge test set, introduced in section 5.1. The TripClick dataset contains real user queries and click-based annotations. It consists of 1.5 million passages and $680k$ training queries. Test queries are divided with respect to their frequency into three sets of $1,750$ queries respectively; the three sets are Head, Torso, and Tail. For the Head queries a DCTR [CMdR15] click model was used to create relevance signals from the click labels. We evaluate on the Head DCTR and the Torso Raw test set as in related work [RLS+21, HASH22, AHVH22].The TripJudge test set contains relevance judgements for the Head test queries, where the relevance judgements are based on human annotations.

Neural ranker details

We train MonoBERT, ColBERT and DPR using training triplets with a RankNet loss [Bur10]. The triplets consist of the query, a relevant and an irrelevant passage; negative passages are taken from the top 1000 BM25 negatives. We train DPR and ColBERT with a batch size of 100, while we use a batch size of 32 for MonoBERT due to its high computational requirements. We train all models for 200 epochs with a learning rate of 7×10^{-6} and we use early stopping. For training, we impose a maximum input length of 30 tokens for the query and 200 tokens for the passage; this setting truncates only a few outlier samples in the dataset but provides computational advantages for batching.

In ❶ **Scratch** we perform fine-tuning from scratch; as underlying PLMs we use DistilBERT [SDCW19] for DPR and ColBERT and the `bert-base-uncased` model [DCLT19] for MonoBERT both provided by Huggingface. We choose these models as starting point so that they match the fine-tuned models for ❷ **Adapt**. In ❷ **Adapt** we start with neural rankers fine-tuned on

MS Marco. For DPR we start from TASB [HLY+21], trained with knowledge distillation and topic-aware sampling; for ColBERT from a ColBERT DistilBERT model trained with knowledge distillation; for MonoBERT from a `bert-base-uncased` model solely trained on MS Marco.

For MonoBERT and ColBERT, we report results in a re-ranking context, i.e. using these neural rankers to re-rank the top 1,000 results retrieved by BM25. For DPR, we instead consider a retrieval setting, where all the collection is scored and then only the top 1,000 are used for evaluation. However, the findings we observe for DPR in the retrieval setting are similar to those we obtained for the same PLM in a re-ranking setting (not reported here). We decided to report retrieval results for DPR, rather than re-ranking as for the other two PLM, because DPR is more commonly used for retrieval (while the other two for re-ranking) [KOM+20, QDL+21, XXL+20].

Active learning details

As foundational experiment we train the neural rankers on different subsets of the training data of differing sizes; as size, we explore the values $[1k, 5k, 10k, 20k, 50k]$. We repeat these experiments 4 times with different random seeds for sampling the subsets, so that each time we train on different subsets with the same size and we can measure variance.

For the active learning process, we increase the training subsets incrementally as denoted in Algorithm 1. In each iteration we train the neural ranker from scratch to exclude a potential bias from incrementally training a ranker. In the first iteration we randomly select the first subset with the same random selection across the different active learning strategies. For uncertainty and diversity selection one could select the first batch with the selection strategy, however for QBC this is not possible different committee members for selection are not available in the first iteration. Therefore we do random selection in the first iteration to be able to fairly compare across the three strategies.

We use random selection as a baseline and also increase the training set incrementally. We run the random baseline 4 times with different random seeds.

For fast and resource efficient active selection, we train the neural rankers for 15 epochs and use the trained ranker for active selection. For the sake of evaluation and in order to compare the effectiveness at different iterations, we resume the training after 200 epochs.

For the uncertainty-selection and QBC strategies, we score the BM25 top 100 passages of each training queries and use these passages for actively selecting the queries for annotation. For the QBC selection strategy we use the same hyper-parameters as Cai et al. [CGZW11]; we use 2 members in the committee and train each member on 80% of the subset available at each iteration for training. We choose the size of the training subsets so that each 80% portion aligns with the other training set sizes.

For ❶ **Scratch** we add in each iteration $s = 5,000$ training samples to the training size. For ❷ **Adapt** we have $s = 5,000$ samples for the first 2 iterations until the training size is 10k, from then on we use $s = 10,000$ samples for the remaining iterations in order to decrease computational cost.

Costs for Budget-Aware Evaluation

For computing the annotation effort, for each triplet added to the training we store the rank of the first relevant document in the ranked list generated by the neural ranker trained in the previous iteration. Since we do not have a trained neural ranker in the first iteration, we start with the initial ranking provided by BM25. For the random baseline, we also use the initial BM25 ranking for computing the number of assessments.

We conduct our experiments on servers equipped with NVIDIA A40 GPUs and measure the GPU and CPU hours spent in the training of the neural ranker and the execution of the selection strategies.

For the computational cost, we refer to common cloud computing costs[5] and set $G_h = 3.060\$$ and $C_h = 0.408\$$. For the number of annotations per hour A_h we rely on estimates from Althammer et al. [AHVH22], who conducted an annotation campaign on TripClick test set. Here annotators needed 47.7 seconds to annotate a query-passage pair on average, which corresponds to 75 assessments per hour. For the annotation cost per hour, A_C, we assume $50US\$$ as hourly rate of a domain expert annotator.

5.2.7 Results

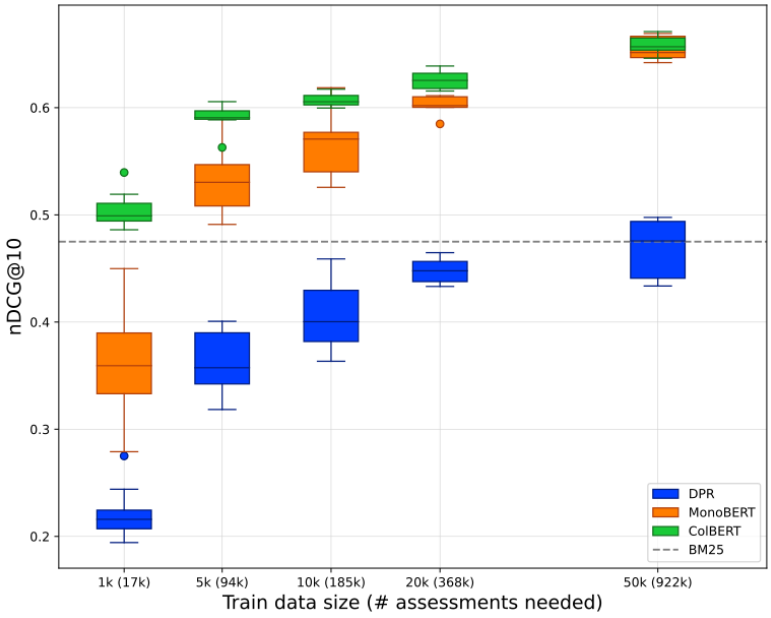
RQ2.2.1: Effect of Size of Training Data

We visualize the effect of training data size on the effectiveness of neural rankers for ❶ **Scratch** (Figure 5.7a) and for ❷ **Adapt** (Figure 5.7b). The boxplots visualize the range of effectiveness when the neural ranker is trained on different subsets of the same size.

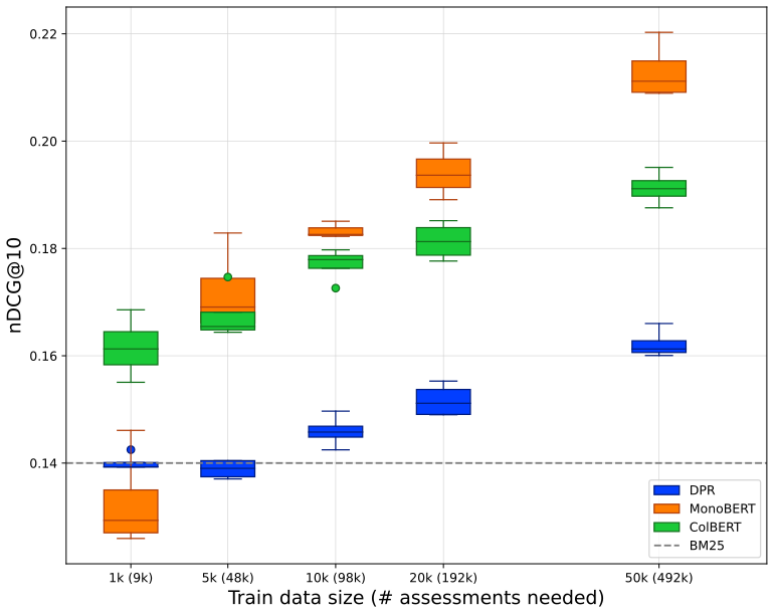
In both cases, it is observed that as the size of the training data increases, nDCG@10 improves for all three neural rankers. When considering effectiveness across neural rankers, it is noteworthy to observe ColBERT and MonoBERT. Recall from the literature that MonoBERT outperforms ColBERT on MS Marco when both are trained on the whole MS Marco training data [CMY+21b], and the same holds for TripClick [HASH22]. However, in our experiments, we observe that ColBERT outperforms MonoBERT for smaller training data sizes. MonoBERT eventually becomes better than ColBERT but only once more than 10,000 training samples are used for the ❷ **Adapt** scenario. For the ❶ **Scratch** scenario the two rankers becomes largely indistinguishable when the training data is 50,000 samples, and eventually MonoBERT takes the lead thereafter (not shown in the figure).

In ❶ **Scratch**, the improvement in effectiveness with increasing training size is particularly remarkable for small training subsets. For example, the improvement by adding 4k training samples from 1k to 5k samples is between 18% and 63% of the median nDCG@10. Noting the wide scale of the y-axis from 0.2 to 0.7 nDCG@10, we observe a large variability when training neural rankers on limited data. This is particularly the case for MonoBERT, where we find a large difference from maximum and minimum nDCG@10 from 19 (0.27–0.46 nDCG@10) for

[5] From `https://aws.amazon.com/ec2/pricing/on-demand/`. Costs valid as of 02 January 2023. GPU costs refer to a `p3.2xlarge` instance and CPU costs to an `a1.4xlarge` instance.



(a) ❶ **Scratch**.



(b) ❷ **Adapt**.

Figure 5.7: Boxplot of nDCG@10 effectiveness on TREC DL 2020 (❶ **Scratch**, Figure 5.7a) and on TripClick Head DCTR test (❷ **Adapt**, Figure 5.7b), visualizing the variability of training on different training sample sizes. neural rankers are trained on subsets of respective sets (MS Marco/TripClick) with different sizes. To measure variability, for each train data size we repeat random sampling 4 times.

Table 5.4: nDCG@10 effectiveness across different amounts of training data for ❶ **Scratch** on TREC DL 2019 & 2020. Bold numbers denote highest effectiveness for each neural ranker and training size. Statistically significant differences to random selection baseline (Random) are denoted with * (paired t-test; $p < 0.05$, Bonferroni correction with n=3). No consistently best performing method and no statistically significant difference to Random. '-' indicates no result at that training size.

nDCG@10		❶ Scratch: MS Marco									
		TREC DL 2019					TREC DL 2020				
Train data size		0	5k	10k	20k	50k	0	5k	10k	20k	50k
0	BM25	.501					.475				
MonoBERT (re-rank BM25 top 1,000)											
1	Random	.051	.5935	.6272	.6430	.6705	.041	.5590	.5871	.6148	.6552
2	QBC		**.6193**	.6157	.6246	**.6728**		.5507	.5844	**.6443**	.6630
4	Uncertainty		.6118	.6232	**.6588**	.6509		**.5875**	.5873	.6336	.6595
5	Diversity		.5925	**.6341**	.6448	.6640		.5407	**.6237**	.6338	**.6670**
ColBERT (re-rank BM25 top 1,000)											
6	Random	.352	.6176	**.6385**	.6352	.6614	.246	.5944	.6091	.6291	.6577
7	QBC		.6192	.6297	**.6541**	**.6680**		.5813	.6159	**.6511**	**.6758**
9	Uncertainty		.6257	.6034	.6089	.6370		**.5987**	.6076	.6001	.6246
10	Diversity		**.6271**	.6239	.6402	.6644		.5912	.6038	.6211	.6363
DPR (full retrieval)											
11	Random	0.0	.3674	**.4390**	.4457	.5006	0.0	.3225	.3789	.4190	.4757
12	QBC		.3465	.4343	.4628	**.5079**		.3023	.3849	.4090	.4534
14	Uncertainty		**.3961**	.4067	.4255	.3757*		**.3660**	.3733	**.4476**	.4254
15	Diversity		.3713	.4086	.4593	.4750		.3437	.4030	.4198	**.4998**

1k samples to 7 (0.53–0.60) for 10k samples. The worst and the best MonoBERT run obtained are statistically significantly different for train size 1k and 5k. For DPR, the inter-quartile range is up to a difference of 5 nDCG@10 (0.38-0.43 for 10k), thus 50% of the effectiveness points are within a range of 5 nDCG@10. A substantial variability in the effectiveness of DPR is observed when trained on 50k samples. The best and the worst runs for DPR are statistically significantly different for 5k and 10k samples. It is noteworthy that the boxplots for 5k samples overlap in part with those for 10k, and similarly the 10k with those for 20k. This means that specific subset of training data of size 5k (10k) allow to reach the same effectiveness obtained when training the ranker on double the amount of data, i.e. 10k (20k).

For ❷ **Adapt** (Figure 5.7b) we also notice variability in search effectiveness; yet, we observe a relatively smaller variability compared to ❶ **Scratch**. The differences between the worst and best runs for each training data size are not statistically significant in this scenario. We suspect that this smaller variability in effectiveness is due to starting from an already fine-tuned neural ranker instead of training from scratch. Although our empirical results suggest a smaller variability, we still see overlaps of the boxplots, especially between 10k and 20k sample: that is, the same or

Table 5.5: nDCG@10 effectiveness across different amounts of training data for ❷ **Adapt** on TripClick Head DCTR & Torso Raw. Bold numbers denote highest effectiveness for each neural ranker and training size. Statistically significant differences to random selection baseline (Random) are denoted with * (paired t-test; $p < 0.05$, Bonferroni correction with n=3). For DPR, Random consistently is best; all statistically significant differences to Random are significantly lower. '-' indicates no result at that training size.

nDCG@10		❷ Adapt: TripClick									
		Head test DCTR					Torso test raw				
	Train data size	0	5k	10k	20k	50k	0	5k	10k	20k	50k
0	BM25	.140					.206				
MonoBERT (re-rank BM25 top 1,000)											
1	Random	.036	.1715	.1833	.1941	.2129	.036	.2279	.2352	.2426	**.2710**
2	QBC		**.1731**	.1835	**.2065**	.2059		.2046	.2328	.2423	.2679
4	Uncertainty		-	**.1920**	.1981	**.2190**		-	**.2356**	.2362	.2705
5	Diversity		-	.1837	.1933	.2123		-	.2294	**.2450**	.2650
ColBERT (re-rank BM25 top 1,000)											
6	Random	.155	**.1675**	.1770	.1813	.1912	.227	**.2300**	**.2351**	**.2397**	.2475
7	QBC		.1302*	**.1791**	**.1860**	**.1962**		.1558*	.2273	.2292	.2360
9	Uncertainty		-	.1645	.1536	.1753*		-	.2190	.1909	.2274
10	Diversity		-	.1645	.1811	.1957		-	.2187	.2362	**.2481**
DPR (full retrieval)											
11	Random	.139	**.1389**	**.1459**	**.1516**	**.1621**	**.200**	**.1837**	**.1745**	**.1924**	**.2023**
12	QBC		.0849*	.1043	.1368	.1603		.0895	.1122	.1312	.1440
14	Uncertainty		.1060	.1165	.1283	.1336		.0907	.0946	.1030	.1031
15	Diversity		.1059	.1150	.1163	.1458		.0907	.1041	.1217	.1473

even better effectiveness could have been reached with half the training data.

These results suggest that it is possible to select subsets of training data that would "speed-up" the learning: in other words, some subsets of training data can achieve the same or even higher effectiveness as using double the amount of data. This thus serves as a motivation for this work: is it possible to identify "*high-yield*" training subsets so as to spare annotation costs but yet obtain high effectiveness? To this aim, we investigate the effectiveness of active learning strategies, which we discuss next.

RQ2.2.2: Effectiveness of Active Selection

We report the effectiveness of the active learning strategies from Section 5.2.4, along with the random selection baseline, when used for training MonoBERT, ColBERT and DPR across different amounts of training data in Table 5.4 for scenario ❶ **Scratch** and in Table 5.5 for ❷ **Adapt**.

For the random selection baseline we report the mean effectiveness when randomly sampling and training on different subsets of the same size multiple times – we perform four random selections

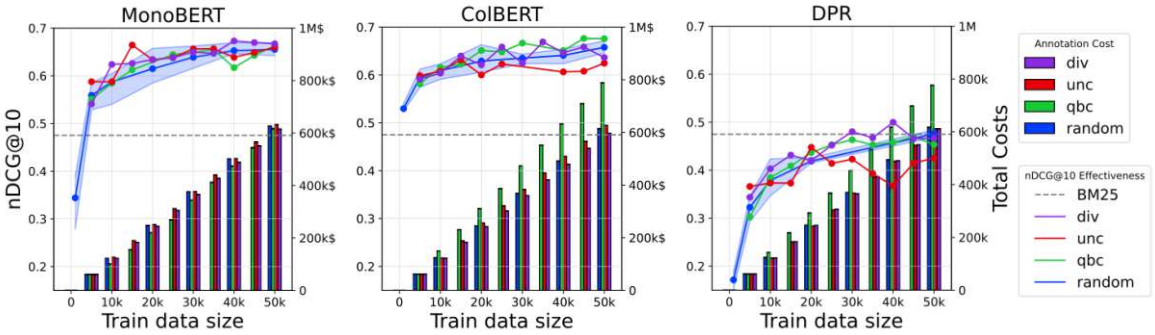


Figure 5.8: nDCG@10 effectiveness (lines, left y-axis) and stacked annotation and computational cost (bars, right y-axis) for different train data sizes on TREC DL 2020 for ❶ **Scratch**. For Random (4 runs) the blue line denotes mean, shaded area denotes the range between min and max effectiveness. Good results would be expected to be between mean and max of Random, bad results between mean and min. For stacked cost, only annotation cost is visible since it greatly exceed computational costs.

for each training size. Note that because the AL selection strategies are deterministic, there is only one result for each strategy at a certain training size, not multiple runs as for the Random baseline.

Various AL strategies outperform the Random baseline at most training data sizes in ❶ **Scratch**. However these effectiveness gains are not consistent throughout all training sizes: there is no single AL strategy that always performs better than the others and, importantly, that always outperforms the random selection baseline. For example, on TREC DL 2020 the uncertainty-based selection for DPR reaches the highest effectiveness when training with 20k samples, but effectiveness drops sensibly when training with 50k samples. Furthermore, effectiveness gains across all methods are not statistically significant, nor are the improvements substantial. When evaluating the neural rankers on MS Marco Dev, we find similar results: there are varying, non-statistically significant improvements of AL strategies to the Random baseline.

The effectiveness results are more consistent across methods and training data sizes in scenario ❷ **Adapt**. Random outperforms all AL selection strategies when using DPR. The QBC strategy reaches slightly higher effectiveness than random selection when ColBERT is used; however, none of the improvements are significant despite the large number of test queries in the TripClick Head and Torso test sets. No statistical significance is found even when the worst random selection run is considered in place of the mean of the random runs.

In summary, we found that for the task of fine-tuning neural rankers, there is no single active learning selection strategy that consistently and significantly delivers higher effectiveness compared to a random selection of the training data. This is a surprising and interesting result. Active learning has been shown to be effective in natural language tasks [LG94], also for methods that rely on PLM models [EDHG+20]: yet, popular AL methods do not work in the context of neural rankers. However, RQ2.2.1 shows that there are subsets of the training data that when used for

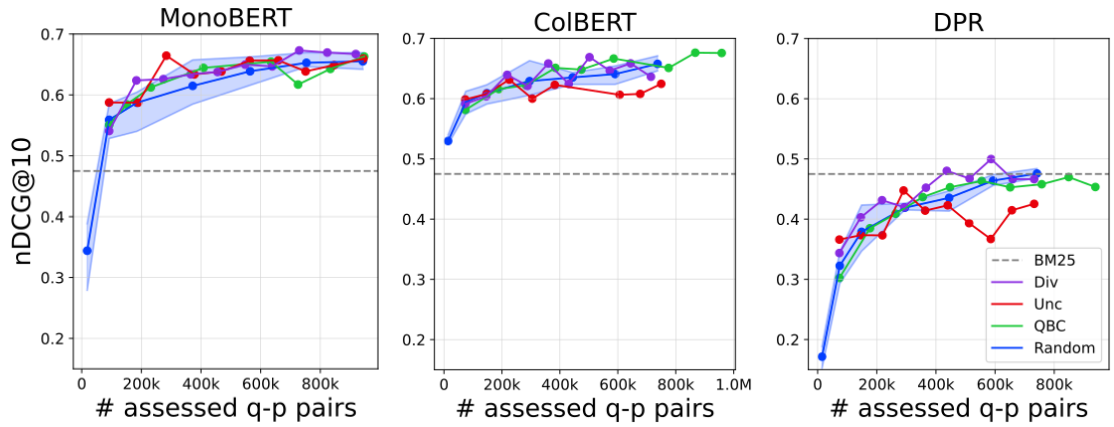


Figure 5.9: nDCG@10 effectiveness versus number of assessed query-passage pairs on TREC DL 2020 for ❶ **Scratch**. Number of assessments per sample is measured with rank of highest relevant passage during selection. For the Random baseline the blue line denotes mean, blue shaded area denotes the range between max and min effectiveness versus mean of the number of assessments. Selection strategies are not consistently more effective considering the number of assessments to annotate the training samples.

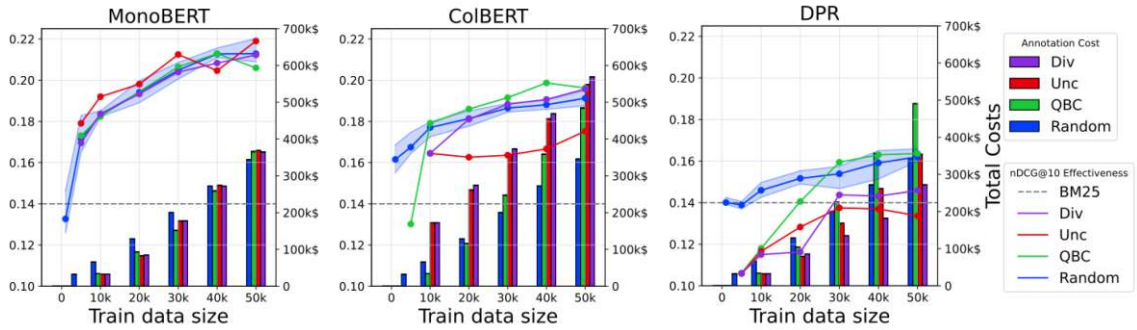


Figure 5.10: nDCG@10 effectiveness (lines, left y-axis) and stacked annotation and computational cost (bars, right y-axis) for different train data sizes on TripClick Head DCTR for ❷ **Adapt**. For Random (4 runs) the blue line denotes mean, shaded area denotes the range between min and max effectiveness. Good results would be expected to be between mean and max of Random, bad results between mean and min. For stacked cost, only annotation cost is visible since it greatly exceed computational costs.

fine-tuning neural rankers deliver sensibly higher effectiveness than others – but AL methods are unable to identify those high-yield training samples.

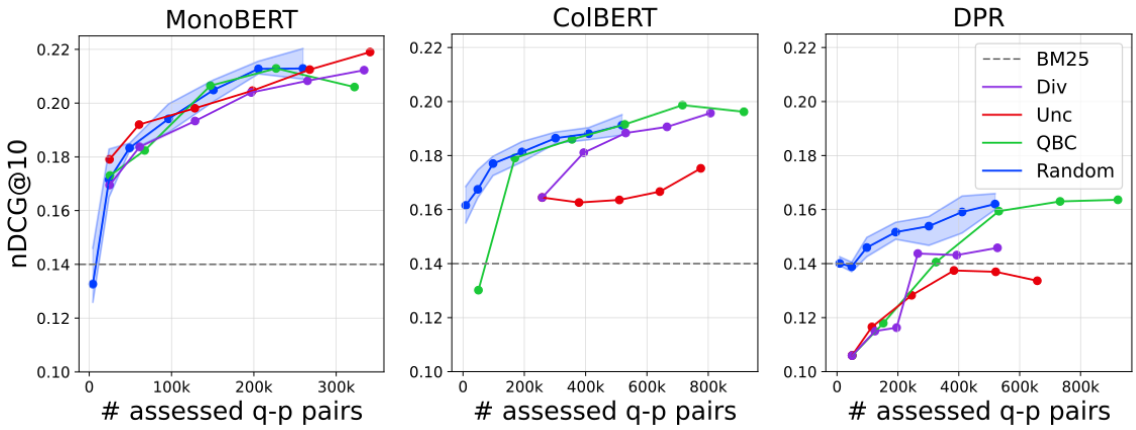


Figure 5.11: nDCG@10 effectiveness versus number of assessed query-passage pairs on TripClick Head DCTR for ❷ **Adapt**. Number of assessments per sample is measured with rank of highest relevant passage during selection. For the Random baseline the blue line denotes mean, blue shaded area denotes the range between max and min effectiveness versus mean of the number of assessments. Selection strategies are not consistently more effective considering the number of assessments to annotate the training samples.

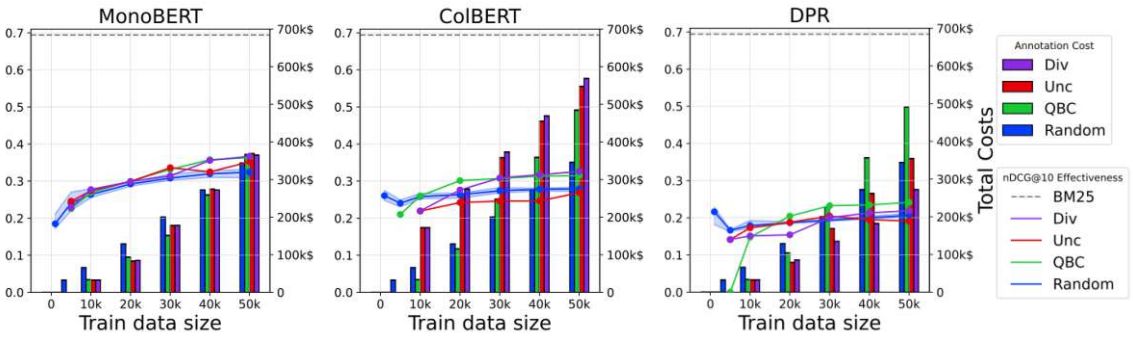


Figure 5.12: nDCG@10 effectiveness (lines, left y-axis) and stacked annotation and computational cost (bars, right y-axis) for different train data sizes on TripJudge for ❷ **Adapt**. For Random (4 runs) the blue line denotes mean, shaded area denotes the range between min and max effectiveness. Good results would be expected to be between mean and max of Random, bad results between mean and min. For stacked cost, only annotation cost is visible since it greatly exceed computational costs.

RQ2.2.3: Budget-aware Evaluation

Since the goal of actively selecting training data is to minimize the annotation cost, we investigate the active selection strategies in the context of constrained budgets. For this, we use the budget-aware evaluation of Section 5.2.4, which accounts for the number of assessments needed to annotate the training data as well as the computational cost of the training and selection.

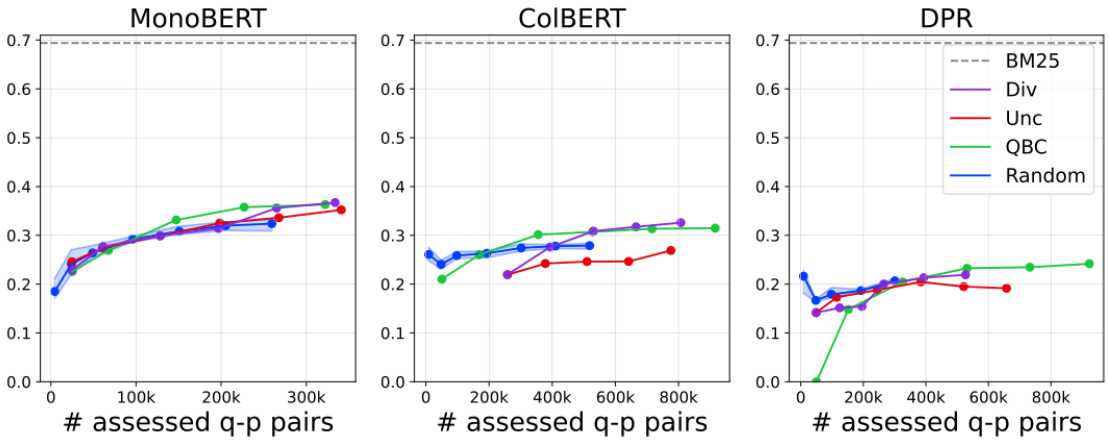


Figure 5.13: nDCG@10 effectiveness versus number of assessed query-passage pairs on TripJudge for ❷ **Adapt**. Number of assessments per sample is measured with rank of highest relevant passage during selection. For the Random baseline the blue line denotes mean, blue shaded area denotes the range between max and min effectiveness versus mean of the number of assessments. Selection strategies are not consistently more effective considering the number of assessments to annotate the training samples.

We visualize the effectiveness and associated costs at different training set sizes for the AL strategies for the three neural rankers in Figure 5.8 for ❶ **Scratch** on TREC DL 2020 and in Figure 5.10 and 5.12 for ❷ **Adapt** on TripClick Head DCTR and TripJudge. The lines and the left y-axis refer to the rankers' effectiveness, measured as nDCG@10. The bars and the right y-axis refer to the total cost computed with the budget-aware evaluation. The bars are stacked (the annotation and computational cost), but since with our cost settings the annotation cost greatly exceeds the computational cost, the bars for the GPU and CPU costs are not visible. In all figures the blue line denotes the effectiveness of Random, with the blue shade representing the range measured between the worst and best random selection runs (recall that random selection was ran four times, and Random is the mean effectiveness of these runs).

A first observation is that the main cost factor is the annotation cost, and hence the number of assessments needed to create the training data, which largely overrules the computational cost. Because of this, in Figures 5.9 (❶ **Scratch**) and 5.11 and 5.13 (❷ **Adapt**) we further visualise the effectiveness of the AL strategies relative to the number of assessments needed to reach that effectiveness.

Next, we analyse the results for ❶ **Scratch** (Figures 5.8 and 5.9). For MonoBERT, the active selection strategies often provide higher effectiveness than Random when more than 10k samples are available – these effectiveness gains are however not significant. Nonetheless, QBC and diversity require a lower budget than the Random baseline with savings of up to 15k$ when 50k query-document samples are collected. We note that the uncertainty-based strategy provides similar effectiveness to Random (especially from 20k samples), at no cost-savings.

For ColBERT, QBC consistently provides higher effectiveness than Random, however at a much higher cost. For example, when 50k training samples are selected, using QBC costs nearly $200k more than Random, requiring annotations for roughly 200k more query-document pairs. In fact, when approximately the same budget/number of annotations are used, QBC and Random obtain the same effectiveness (in Figure 5.9 compare the last point of Random with the third last point of QBC). Aside from QBC, all other active selection strategies deliver similar or lower effectiveness of Random, for the same or higher cost.

For DPR, the uncertainty-based strategy consistently delivers inferior effectiveness than the baseline. QBC and diversity-based selection do provide effectiveness gains when the training data is in the range 10k to 40-45k samples. For QBC, however, these gains come at a large budget expense: for 30k the QBC selection requires 90k$ more annotation budget than Random. The diversity-based strategy instead does deliver some costs-savings compared to Random. For example, for Random to reach the same effectiveness of BM25, about 600k annotations are needed, while diversity delivers the same level of effectiveness with only 420k annotations. However, we note using more annotations with diversity-based sampling does not necessarily translate in a more effective model: going from 600k to about 750k annotations deteriorates the search effectiveness of the ranker.

Looking across neural rankers, we observe that while the annotation costs across selection strategies are relatively similar for MonoBERT, they are higher for QBC than all other strategies when ColBERT and DPR are considered.

Overall, the selection strategies show relatively unstable effectiveness, the effectiveness can even decrease when training data increases. This is particularly the case for uncertainty selection for DPR: for example, its effectiveness decreases by from 0.45 to 0.37 when the amount of training data doubles from 20k to 40k.

We next analyse the results for scenario ❷ **Adapt**. While some selection strategies provide gains over random selection, these gains largely depend on which PLM is used and the training size (Figure 5.10 and 5.12). Nevertheless, despite the specific gains in effectiveness, all active selection strategies require more assessments, and thus a higher budget, to reach the same level of effectiveness obtained when using random selection (Figure 5.11 and 5.13).

On TripClick test, for MonoBERT, uncertainty selection exhibits (non-significant) improvements when training data is less than 30k. In fact, for small amounts of training data, uncertainty sampling does provide some cost savings: for example MonoBERT with uncertainty sampling needs about 65,000 query-passage pairs assessments to obtain the same effectiveness obtained with random selection with $\approx 100k$ assessed pairs. However, this effect is lost when the training data size increases further, with the budget required by uncertainty sampling becoming similar (or more in some instances of random selection) to that of Random to obtain the same level of effectiveness. All other active selection strategies, when used with MonoBERT, deliver either lower effectiveness than Random, or higher costs. This is the case particularly for QBC. In fact, although there is one setting in which QBC delivers major cost savings to reach the same effectiveness of the random baseline (QBC achieves nDCG@10 higher than 0.2 using a sensibly lower amount of annotated query-passage pairs), cost savings are not consistent across all training

data sizes and larger sizes correspond to a higher number of assessments required compared to Random.

For ColBERT, QBC and diversity selection outperform the baseline from training data sizes of 20k onward. This however comes with a considerable increase of query-passage pairs to be assessed and thus of annotation cost. For example, with a training subset of 30k, random selection costs about $200,000 while diversity selection costs nearly double that – but the increase in search effectiveness is marginal. It is interesting to compare these results with that obtained for scenario ❶ **Scratch**. While in both scenarios uncertainty selection shows effectiveness losses when more training data is added, and QBC is associated with higher costs, diversity selection performs differently: it provides similar effectiveness for a similar cost in ❶ **Scratch**, and a marginal effectiveness improvement for a largely higher cost in ❷ **Adapt**.

For DPR, all selection strategies underperform random selection, with the exception of QBC that provides marginal improvements when the training subset is larger than 30k, but this at the expense of a higher budget. The budget-aware evaluation, in fact, shows that all selection strategies require more query-passage pairs to be assessed (higher cost) than the random selection baseline to reach the same search effectiveness (and some strategies cannot even achieve that effectiveness). An example is QBC that requires 730,000 assessments to reach the same effectiveness obtained by Random with just $\approx 250k$ assessments.

We see a similar picture when evaluating scenario ❷ **Adapt** with TripJudge. Overall all neural rankers greatly underperform BM25 in terms of effectiveness, however this is already observed when using the TripClick training data based on user clicks, but evaluating on TripJudge [AHVH22]. For MonoBERT, all selection strategies improve over random selection when trained on 40k samples or more. However the active selection strategies require more assessments than random selection. For ColBERT, we see the similar picture as for TripClick Head DCTR: QBC and diversity-based selection outperform random selection at a certain training size, whereas uncertainty-based selection has a negative impact on the effectiveness of the neural ranker. For DPR only QBC selection outperforms random selection, however the active selection strategies require largely more assessments for training DPR than random selection. Additionally the effectiveness gains on TripJudge are not significant for all neural rankers.

In summary, in answer to RQ2.2.3, we found that the use of the investigated active selection strategies does not deliver consistent budget savings. In our experiments, the budget is largely dominated by the assessment cost and all active selection strategies tend to require a higher amount of query-passage pairs to be annotated than random selection. Even in contexts where assessment is very cheap, active selection would not provide budget savings because more assessments are required for active selection than for random selection. We do note that there are cases where specific active selection strategies provide similar search effectiveness than random selection at a reduced cost. However, these cases occur for specific choices of selection strategy, neural ranker and training subset size and thus are unlikely to generalise in practice.

5.2.8 Conclusion

We investigated fine-tuning neural rankers under limited data and budget. For this, we adapted several active selection strategies, representing different key approaches in active learning that have been shown effective in many natural language processing tasks. Surprisingly, we found that for the task of fine-tuning neural rankers no AL strategy consistently and significantly outperformed random selection of training data. However we found that there are subsets of the training data which lead to significantly higher effectiveness than others, thus we see it as an important open challenge to be able to automatically identify those training samples. Similarly, our budget-aware evaluation showed that the investigated AL strategies do not deliver consistent budget savings since they require a higher amount of assessments than random selection.

Limitations and future work

One limitation of our study is that the estimation of annotation costs relies on sparse annotations of the training set. Potentially, the required number of assessments could be lower, since another relevant passage – that is not marked as relevant in the data – could be found earlier in the ranked list.

Furthermore another limitation that is related to the measurement of annotation cost, is that we only simulate the selection and annotation process, but in reality we have a static collection with a set of static, already determined labels, before we start the process. This could also lead to differences in the selection of training samples. We argue, however, that this should affect all selection strategies and does not benefit one strategy particularly. Possible future experiments including a real annotation pipeline during training where it is possible to measure the annotation costs of annotating each sample would be needed to test the if the findings of this study also hold with a live annotation campaign and with the real measured cost of annotations.

Another limitation is the way uncertainty was computed in our experiments. Uncertainty estimation in Information Retrieval is a fundamental but largely unexplored problem [TC96, CLvR98, CC07], especially for neural rankers [LRC+21, CML+21]. Attempts have been made to exploit uncertainty in relevance estimation for traditional statistical models such as language models and BM25 [ZWCT09, WZ09], but in these works the actual estimation of uncertainty is based on assumptions and heuristics such as to be related to similarities or covariance between term occurrences [ZWCT09, WZ09, ZA10], to follow the Dirichlet distribution [WZ09], or to be computed based on score distributions obtained through query term re-sampling [CC07]. Recent attempts have been made to model uncertainty for neural rankers, for example Transformer Pointer Generator Network (T-PGN) model [LRC+21], or Cohen et al.'s [CML+21] efficient uncertainty and calibration modelling strategies based on Monte-Carlo drop-out [GG16], but these are not readily applicable to the neural ranker architectures we consider. In future work we plan to adapt and investigate these uncertainty estimations.

Another limitations is the choice of tasks and datasets. While this study presents results on MS Marco and TripClick, it is an open question how our findings generalize to other retrieval and ranking tasks and datasets.

Finally we also highlight that we only considered common baseline active learning methods

[XAZ07, LG94, FSST97]. More sophisticated AL methods exist [AZK+20, YLB20, MVBA21], including alternating between selection types like in AcTune, which alternates active learning and self-training [YKZ+22], and Augmented SBERT which alternates random selection and kernel density estimation based selection [TRDG21]. However, each of these approaches present specific challenges to be adapted to ranking. We also were interested to understand the promise AL has for neural rankers, and provide a framework, inclusive of evaluation methodologies and baselines, in which these more advanced methods could be studied.

CHAPTER 6

Conclusion

In this thesis, we investigate how neural ranking and retrieval models can be adapted for document-to-document retrieval tasks and we address the problem of data availability for evaluation and training of neural ranking and retrieval models in domain-specific retrieval.

In the beginning of the thesis we introduced **Search Example ❶**, which is a legal attorney, who needs to find related prior cases to his current case. In **Search Example ❷** a patent attorney needs to find prior patents that are related to his new patent application and in **Search Example ❸** a medical doctor needs to find treatment strategies for a patient in the emergency department. In this thesis and in the context of the research questions, we have addressed the challenges of neural ranking and retrieval models for the retrieval tasks in **Search Example ❶-❸**. **Search Example ❶** and **Search Example ❷** are examples for the document-to-document retrieval tasks of prior case retrieval in the legal domain and prior art search in the patent domain and in this thesis we have studied how we can adapt neural ranking and retrieval models for prior case retrieval and prior art search. For medical ad-hoc retrieval, exemplified in the **Search Example ❸**, we have addressed in this thesis the problem of limited evaluation and training data, when training neural ranking and retrieval models. We have conducted a human annotation campaign, in order to create TripJudge, a test collection for the task of medical ad-hoc retrieval. Furthermore we have investigated how to train neural ranking and retrieval models for ad-hoc retrieval in the medical and web domain under a limited annotation and training budget.

6.1 Revisiting Research Questions

We revisit the research questions introduced in section 1.2 and lay out how we addressed the research questions in this work.

6.1.1 Domain-specific neural rankers for document-to-document retrieval tasks

Since neural ranking and retrieval models show great effectiveness gains for ranking and retrieval tasks in the web domain, it is an open and important question how these findings generalize to domain-specific ranking and retrieval tasks. Especially for document-to-document retrieval tasks with long queries and long documents, it is not trivial and an open question how neural ranking and retrieval models can be adapted to these document-to-document retrieval tasks. We tackle the research question:

RQ1 How can neural ranking and retrieval models be adapted for document-to-document retrieval tasks?

We divide this research question between neural ranking and neural retrieval models and investigate in section 4.1 the following research question:

RQ1.1 How can neural ranking models be adapted for document-to-document retrieval tasks?

In order to study how neural ranking models can be adapted to document-to-document retrieval tasks, we successfully reproduce Shao et al.'s [SML+20] BERT-PLI model for the legal document retrieval task evaluated in the COLIEE evaluation campaign 2019. In doing so, we address certain shortcomings in the data pre-processing. Our investigation led us to complement the published code. However, in contrast to the original paper, our findings suggest that fine-tuning a BERT model on domain-specific data for modeling paragraph-level interactions does not significantly enhance the performance of the BERT-PLI model for document re-ranking when compared to using the original BERT model for this purpose.

Additionally, we explore the applicability of the BERT-PLI model in the patent domain for the task of prior art search but find that it does not outperform the BM25 baseline. Yet, effectively harnessing the potential of contextualized language models for patent document re-ranking remains an unsolved challenge.

We also investigate the transferability of the BERT-PLI model between the legal and patent domains, both at the paragraph and document level in the BERT-PLI model. Our results demonstrate comparable performance when transferring the model at the paragraph level. Moreover, initial results in cross-domain document-level transfer indicate promise when applying a BERT-PLI model trained on the patent domain to the legal domain. The question of how to transfer the concept of relevance across these domains remains intriguing and open.

Additionally to investigating the adaptation of neural ranking models for document-to-document retrieval tasks, we investigate:

RQ1.2 How can neural retrieval models be adapted for document-to-document retrieval tasks?

We study how neural first stage retrieval models can be adapted for document-to-document retrieval tasks or legal case retrieval and prior art search. In this work we address the challenges of using dense passage retrieval models (DPR) in first stage retrieval for document-to-document tasks when training the dense passage retrieval model with limited labelled data.

Our solution, the Paragraph Aggregation Retrieval Model (PARM), overcomes the constraints of input length in dense passage retrieval models and takes into consideration the relevance of paragraphs in the context of document retrieval. When applied to legal case retrieval using two test collections, PARM demonstrates a higher first-stage recall in dense document-to-document retrieval compared to document-level retrieval with the fixed input length DPR. Moreover, in the case of legal case retrieval, we show that dense retrieval with PARM surpasses lexical retrieval using BM25 in terms of recall at higher cut-off values.

As an integral component of PARM, we introduce the novel Vector-Based Aggregation with Reciprocal Rank Fusion Weighting (VRFF). VRFF combines the benefits of rank-based aggregation using Reciprocal Rank Fusion (RRF) and topical aggregation with dense embeddings. Our experiments reveal that PARM with VRFF aggregation achieves the most effective retrieval performance when compared to rank and vector-based aggregation baselines.

However, in the context of prior art search for document-to-document retrieval, we find that dense retrieval methods based on PARM and standard dense retrieval techniques do not outperform lexical retrieval using BM25.

To ensure the reliability of evaluation results on the CLEF-IP test collection, we suggest future work to investigate the suitability of this test collection for evaluating neural first-stage retrieval models, given that only statistical models were involved in its creation.

Additionally, we delve into the training of dense retrieval models for dense document-to-document retrieval with PARM. Notably, in the realm of legal case retrieval, our findings indicate that training DPR models on more but noisy document-level data doesn't consistently result in higher overall retrieval performance compared to training on less but more accurate paragraph-level labeled data.

Finally, we conduct an analysis of how PARM retrieves relevant paragraphs and observe that the dense retrieval model learns a structural paragraph relationship, which it utilizes to enhance retrieval effectiveness in the context of PARM.

Overall answering research question **RQ1**, we successfully and effectively adapt neural ranking and retrieval models for the task of prior case retrieval in the legal domain. We evaluate the neural models in the context of the tasks' requirements, that is a high recall for the first stage retrieval and a high precision for re-ranking. We find that domain-specific neural re-ranking and first stage retrieval models that take into account the whole content of the query document and the document in the collection, are beneficial for the effectiveness of the model. However for prior art search in the patent domain, we find that domain-specific neural ranking and retrieval models do not yet bring the expected effectiveness gains compared to traditional, lexical ranking models.

6.1.2 Availability of evaluation and training data for domain-specific tasks

A second great, open challenge of ranking and retrieval models for domain-specific tasks is the availability to training and evaluation datasets. Thus we investigate in this thesis:

RQ2 How can the problem of limited available annotated evaluation and training data be addressed in domain-specific retrieval?

We address the problem of limited available evaluation data by providing the community with a human-label annotation test collection for health retrieval. When proposing this novel test set, we investigate:

RQ2.1 How do human-label annotations compare to click signals for medical ad-hoc retrieval?

We address the limited availability of annotated evaluation data for domain-specific retrieval tasks by successfully running an annotation campaign for an ad-hoc retrieval task in the health domain. Here we compare our human-label annotations with the original relevance labels based on click signals. We find that human relevance annotations greatly differ from the click-based relevance labels and thus the ranking of lexical and neural ranking and retrieval models highly differs between the two different test sets.

Furthermore the domain-specific neural ranking and retrieval models do not as greatly outperform the lexical retrieval baseline, when evaluated on the humanly judged evaluation set, compared to the great effectiveness gains on the click-based test set. We suggest that by training on the click-based labels, the neural ranking and retrieval models learn the relevance signals from the clicks, thus the effectiveness improvements on the click-based test collection are larger than on the human-labelled test set and potentially exaggerated. This demonstrates that high effectiveness of domain-specific neural ranking and retrieval models needs high-quality training data.

This finding leads us to the next research question, where we investigate how we can train neural ranking and retrieval models under a limited training data annotation and model training budget. Here active learning strategies are a promising direction for minimizing the amount of annotations of training data while maximizing the effectiveness of the models trained on that training data. Here we investigate:

RQ2.2 To what extent does active learning improve annotation efficiency for training neural ranking and retrieval models?

Since it is costly to annotate domain-specific training data on a large scale, we study to what extent active learning methods improve the annotation efficiency for training effective neural ranking and retrieval models. In order to have no potential influence of domain-specific neural ranking and retrieval architectures, which we proposed in chapter 4, we study this research question in the context of ranking and retrieval tasks in the web and health domain. We see varying gains of the investigated active selection strategies compared to random selection, but with our cost-effective evaluation schema we find that these gains come at the cost of more assessments for annotating the training samples. Thus the investigated active selection strategies do not yet minimize the

annotation cost while maintaining a high effectiveness. However we find that there subsets of high yield training samples that achieve significant effectiveness improvements compared to random selection, but the investigated active learning strategies do not identify them. Furthermore we find that adapting neural ranking and retrieval models from the web to the health domain already outperforms statistical retrieval models, when adapted only on small subsets of the training data. Overall we find that neural ranking and retrieval models are beneficial for ad-hoc retrieval in the web and health domains and outperform strong lexical retrieval baselines, when enough, high-quality training data is available.

Overall we find that domain-specific neural ranking and retrieval models advance the performance for domain-specific retrieval tasks, when adapting them to the tasks' specific characteristics and having reliable evaluation and enough high-quality training data available.

6.2 Limitations

We want to conclude limitations of our work and summarize the limitations of the individual studies already described in sections 4.1.5, 4.2.5,5.1.7, 5.2.8.

Neural rankers for document-to-document retrieval tasks

For neural ranking and retrieval models, that we studied in the context of document-to-document tasks, we see a limitation of our study that we focus only on English text of documents. While legal case retrieval and prior art search are important and often studied tasks in English [RGK+22, NFIH10], these tasks also exist in contexts where the documents are in other languages than English [MSW+21, PLHZ11] or the documents contain multiple languages [PLHZ11]. In this work we have only studied the neural re-ranking and neural retrieval models in English due to the availability of training and evaluation datasets in English. However it remains open, how these findings generalize to document-to-document retrieval tasks with languages other than English.

Another limitation pertains to the generalizability of the findings regarding neural ranking and retrieval models when applied to document-to-document retrieval tasks involving documents without a predefined structure. We have studied neural ranking and retrieval models in the context of legal case retrieval and prior art search. Here the legal claims and the patents have a through, predefined structure, that organizes the long documents into topically coherent sections. The neural ranking and retrieval models leverage this predefined structure and operate on a paragraph-level, thus how the documents are split up into different paragraphs is an important factor for the neural ranking and retrieval models. As in legal case retrieval, the relevance of a document can be determined of relevance of paragraphs of the query document and the document itself [RKG+20], the neural ranking and retrieval models use that to their advantage. Similarly in prior art search, the patent documents consist of topically very different sections and relevance can be determined on the relevance between those sections. However, it remains unclear and necessitates further research to determine how the insights derived from neural ranking and retrieval models for legal case retrieval and prior art search apply for document-to-document retrieval tasks, where paragraph divisions are not predefined by semantically coherent sections in a structured document.

Annotation campaign for addressing availability of evaluation datasets

For addressing the problem of data availability for evaluation and training in domain-specific retrieval tasks, we conduct an human-annotation campaign and create TripJudge, a test collection for TripClick based on human-annotation. Our study on TripJudge has several limitations. First, the depth of relevance judgments is typically only a half or a third of the annotated query-document pairs, compared to our 65% relevance rate [CMYC19, CMY+21b]. This suggests a need for future work involving higher-depth annotations.

Another limitation is that TripJudge relies on annotations from non-experts, and not the entire test set is annotated by experts. To control the quality of non-expert annotations, we conducted an additional expert annotation campaign to validate the non-expert annotations. While there was a high overlap between expert and non-expert annotations, the reliance on non-expert annotators remains a limitation of the study.

Additionally, the number of annotators per sample is constrained. We had to strike a balance between annotating all test set samples and the number of annotators available, leading to our decision to have three annotators per query-document sample. Having more annotators per sample would enhance the quality of the test set.

Moreover, this study did not explore various methods for aggregating annotations per query-document sample. While majority voting was employed, future work could investigate more sophisticated heuristics for label aggregation.

Nevertheless, TripJudge remains a valuable resource for evaluating domain-specific health retrieval tasks.

Active learning for training neural rankers

When investigating in Section 5.2, how we can train neural ranking and retrieval models, we conduct experiments on the three model architectures introduced in Section 2.6, however it remains open how the findings would translate to other ranking model architectures like [ZQJ+23].

Another limitation of this study is that we only simulate the selection and annotation process, while in reality, we have a static training collection with predetermined labels before we begin. This could introduce variations in the selection of training samples. However, we argue that this should impact all selection strategies uniformly, rather than favoring one strategy. Furthermore a limitation of the simulated annotation process, is that we can not measure the annotation cost of annotating a query-document sample. The estimation of annotation costs in our study relies on sparse annotations of the training set. It is possible that a lower number of assessments might suffice, as an additional relevant passage not marked as such in the data could be found earlier in the ranked list. To address this limitation, future experiments involving a live annotation campaign, with the actual measured costs of annotations, would be necessary to validate the findings of our study.

Another limitation relates to how uncertainty was computed in our experiments. Estimating uncertainty in Information Retrieval is a fundamental yet underexplored issue, especially for rankers based on pre-trained language models. While some attempts have been made to exploit

uncertainty in relevance estimation, they often rely on assumptions and heuristics, such as similarities or covariance between term occurrences [CC07]. Recent attempts to model uncertainty for neural rankers might not be readily applicable to the architectures we considered. Future work will involve adapting and investigating these uncertainty estimation methods.

Lastly, our study only considered common baseline active learning methods [CGJ96, XAZ07, SOS92, FSST97]. More sophisticated active learning methods exist, including those that alternate between selection types. These approaches present specific challenges when adapting them to ranking tasks. Our intention was to provide a framework and evaluation methodologies for studying the promise of active learning in neural rankers, and more advanced methods can be explored in this context.

Additionally, our study focuses on specific tasks and datasets, namely MS Marco and TripClick. Generalizing our findings to other retrieval and ranking tasks and datasets remains an open question.

Overall limitations of the thesis

Overall there are limitations of this thesis concerning the choice of language model as encoder, choice of model architectures, choice of datasets, choice of tasks and choice of domains for domain-specific neural rankers.

There exist a lot of different large, pre-trained language models [LOG+19, LCG+20, SDCW19, RSR+20, BMR+20, DCLT19] that could be used as encoder language model base for the neural ranking and retrieval models. Since BERT and DistilBERT are the most common language models used as encoders [HLY+21, NC19, NYCL19] for neural ranking and retrieval models, we focus in our thesis on these language models as encoders. We employ domain-specific language models like LegalBERT [CFM+20] for encoding domain-specific language, however these models have the same architecture as BERT and are only pre-trained on a different language corpus. Thus a limitation of our thesis is the choice of large, pre-trained language model as encoder and the effect of different encoder models for domain-specific neural rankers is an open question.

Furthermore a limitation of this thesis is the choice of model architectures. We study the neural re-ranking model architectures of cross-encoder BERT (MonoBERT) and ColBERT, which we introduced in section 2.6. However there are other model architectures like RankT5 [ZQJ+23], who employ a different language model and a different ranking model architecture as neural re-ranking model. The generalizability of our findings to other neural model architectures remains an open research question.

In our experiments we limit our thesis to the training and evaluation datasets, which we introduced in Section 2.5. For example for the legal case retrieval task it remains open, how BERT-PLI or PARM would perform for the FIRE AILA dataset, which is a dataset for Indian case law retrieval established in the FIRE evaluation campaign [BGG+19]. As already mentioned above, for the investigation of active learning for neural ranking and retrieval models, it remains open how active learning strategies would have performed for training neural ranking and retrieval models on other training and evaluation datasets.

Another limitation of our thesis is the choice of tasks for our experiments. In section 2.4 we have introduced various ranking and retrieval tasks in the web, legal, health and patent domain. However we limit us in our experiments to the tasks of legal case retrieval in the legal domain, patent prior art search in the patent domain, health information seeking/ad-hoc retrieval in the health, and ad-hoc retrieval in the web domain. Especially when investigating how to train neural ranking and retrieval models under a limited training and annotation budget, it is a limitation to investigate ad-hoc retrieval tasks and it is an interesting question how active learning strategies perform for training neural rankers for other ranking and retrieval tasks.

Finally a limitation of this thesis is the choice of domains, in which context we conduct our experiments. We employ tasks in the web, legal, health and patent domain, however there are many other domains including domain-specific and professional information seeking tasks like news domain [SHH20], music domain [KJC+21], cooking domain [FEL22], where neural ranking and retrieval models can be explored and where these models also promise effectiveness gains in performance. Thus it is an open question how the findings of this thesis generalize to other domains and other ranking and retrieval tasks in these domains.

6.3 Future Work

In the following we point out some possible future these directions and some challenging, open questions to be addressed for domain-specific, neural ranking and retrieval models. While our research on domain specific document-to-document retrieval tasks and on domain-specific data availability already has sparked some follow up work [AVA22, AVA+23, BGPG22, Sta22, SGHS22, LRA23], there are yet unresolved challenges and novel questions for neural ranking and retrieval models in the context of domain-specific retrieval tasks. Our research lays the basis for multiple future directions for domain-specific neural ranking and retrieval models, which we elaborate in the following.

Large-scale, domain-specific training and evaluation data for document-to-document retrieval

In order to gain a comprehensive understanding of the potential of neural ranking and retrieval models for domain-specific document-to-document retrieval tasks, it is a necessary requirement to have large-scale, high-quality training data and reliable and reusable evaluation sets that are suitable for evaluating neural ranking and retrieval models. For many of domain-specific tasks, we do not yet have the resources to train and reliably evaluate domain-specific neural models as we find with our work on prior art search in the patent domain. For ad-hoc retrieval in the health domain we see a future necessity to have a large-scale, human-labelled, training data available that models the domain-specific notion of relevance. Thus we also see continuous ongoing efforts for creating artifacts for domain-specific evaluation and training as crucial for the advancements in this area. In order to be able to evaluate newly upcoming, domain-specific neural ranking and retrieval models, we see it as an on-going, continuous necessity to have domain-specific evaluation campaigns that include runs of a variety of non-neural and neural ranking and retrieval approaches for judgement.

Generation of domain-specific training data

One potential future direction for advancing the availability of training data is employing the power of large, pre-trained language models for generating training samples [DZM+23] for domains, where no or limited humanly labelled training samples are available. Since these large language models rapidly progress in their capabilities [Ope23], the question of how to employ them to generate high-quality training data becomes more critical. When training domain-specific rankers on generated training data, it is an essential step for a high effectiveness to select the high-quality training samples from the generated training set and only train on those. This selection step is crucial for a high effectiveness of the resulting neural ranking or retrieval model [DZM+23], thus research on how to identify those high-yield training samples is highly relevant also for generating training samples.

Active learning for neural ranking and retrieval

Furthermore we see some future work in extending active learning strategies for neural ranking and retrieval models. Especially how uncertainty of neural ranking and retrieval models is measured is an open but for numerous applications important question [TC96, CLvR98, CC07, LRC+21].

Domain-specific features included in neural ranking and retrieval models

We see one major direction for improving the effectiveness of domain-specific ranking and retrieval models to include features other than the textual data for relevance ranking. Domain-specific features like recency, citations of the document [Wig23], authors of document, date of the document, location, or classifications like MeSH terms in the medical domain or IPC terms in the patent domain highly influence relevance in domain-specific retrieval. Thus including these domain-specific features in the training and inference of neural ranking and retrieval models holds a great potential for boosting the performance of domain-specific neural models, which so far only rely on textual data.

Continually updating neural ranking and retrieval models

For search indices in production, there are continuously millions of new data points which need to be included in the search index in real-time [Sta21, BP98]. To be able to rank or retrieve high-quality, relevant, and recent results, the novel content not only needs to be included in the search index, but the ranking or retrieval models needs to account for the content shift and update the index in real-time. This poses a challenge for neural ranking and retrieval models which are so far trained on a static training collection and it is not clear how neural ranking and retrieval models cope with the temporal evolution of real Web data [GDS+23]. Thus it is an open research direction how neural ranking and retrieval models can continually be updated for ranking or retrieving the novel content [Alt21].

Generative search engines

With the emergence or large foundation models like ChatGPT [Ope23], a more conversational way of information seeking of the users [ZTDR23] and the shift of search engines towards question-answering [Cla18a] in the web domain, it is an open research question, how this movement will influence, how users interact with information retrieval systems and which information needs

they will have in the future. This different way of interacting and searching for information, could also translate to domain-specific retrieval systems. Thus the users could expect domain-specific retrieval systems to offer similar capabilities as web search engines for interacting and searching for relevant information in a more conversational way with a generated response text or a generated answer. For domain-specific retrieval systems this poses challenges, how to adapt the existing models for question-answering [Cla18a], retrieval-augmented generation [LPP+20] or conversational search [ZTDR23] to domain-specific retrieval tasks and thus will be an exciting and promising research direction.

List of Figures

List of Tables

Bibliography

[AAZ18] François Elvinger Loren Rees Weiguo Fan Abdullah Awaysheh, Jeffrey Wilcke and Kurt Zimmerman. A review of medical terminology standards and structured reporting. *Journal of Veterinary Diagnostic Investigation*, 30(1):17–25, 2018.

[ABD06] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 19–26. Association for Computing Machinery, 2006.

[ABHH21] Sophia Althammer, Mark Buckley, Sebastian Hofstätter, and Allan Hanbury. Linguistically informed masking for representation learning in the patent domain. In *Proceedings of the 2nd Workshop on Patent Text Mining and Semantic Technologies (PatentSemTech) 2021 co-located with the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. CEUR Workshop Proceedings, 2021.

[AEAK12] Ahmet Aker, Mahmoud El-Haj, M-Dyaa Albakour, and Udo Kruschwitz. Assessing crowdsourcing quality through objective tasks. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012*, pages 1456–1461. European Language Resources Association (ELRA), 2012.

[AHH21] Sophia Althammer, Sebastian Hofstätter, and Allan Hanbury. Cross-domain retrieval in the legal and patent domains: a reproducibility study. In *Advances in Information Retrieval, 43rd European Conference on IR Research, ECIR 2021*, pages 3–17. Springer, 2021.

[AHS+22] Sophia Althammer, Sebastian Hofstätter, Mete Sertkan, Suzan Verberne, and Allan Hanbury. Paragraph aggregation retrieval model (parm) for dense document-to-document retrieval. In *Advances in Information Retrieval, 44rd European Conference on IR Research, ECIR 2022*, pages 19–34. Springer, 2022.

[AHVH22] Sophia Althammer, Sebastian Hofstätter, Suzan Verberne, and Allan Hanbury. Tripjudge: A relevance judgement test collection for tripclick health retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pages 3801–3805. Association for Computing Machinery, 2022.

[AKF13] Azhar Alhindi, Udo Kruschwitz, and Chris Fox. Site search using profile-based document summarisation. In *Proceedings of the 13th Dutch-Belgian Workshop on Information Retrieval*, volume 986 of *CEUR Workshop Proceedings*, pages 62–63. CEUR-WS.org, 2013.

[AKTVJ01] Khalid Al-Kofahi, Alex Tyrrell, Arun Vachher, and Peter Jackson. A Machine Learning Approach to Prior Case Retrieval. In *Proceedings of the 8th International Conference on Artificial Intelligence and Law*, pages 88–93. Association for Computing Machinery, 2001.

[AL11] Omar Alonso and Matthew Lease. Crowdsourcing for information retrieval: principles, methods, and applications. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011*, pages 1299–1300. Association of Computing Machinery, 2011.

[Alt20] Sophia Althammer. Cross-domain Retrieval in the Legal and PatentDomain: a Reproducability Study. `https://doi.org/10.5281/zenodo.4088010`, 2020. Accessed on Zenodo: 2023-11-01.

[Alt21] Sophia Althammer. Rudi: Real-time learning to update dense retrieval indices. In *Proceedings of DESIRES 2021 – 2nd International Conference on Design of Experimental Search Information REtrieval Systems (2021)*, volume 2950 of *CEUR Workshop Proceedings*, pages 173–175. CEUR-WS.org, 2021.

[AM12] Omar Alonso and Stefano Mizzaro. Using crowdsourcing for trec relevance assessment. *Information Processing & Management*, 48:1053–1066, 2012.

[AOC18] Qingyao Ai, Brendan O'Connor, and W. Bruce Croft. A neural passage model for ad-hoc document retrieval. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*, volume 10772 of *Lecture Notes in Computer Science*, pages 537–543. Springer, 2018.

[Ash14] Kevin D. Ashley. Applying argument extraction to improve legal information retrieval. In *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing, Forlì-Cesena, Italy, July 21-25, 2014*, volume 1341 of *CEUR Workshop Proceedings*, pages 17–25. CEUR-WS.org, 2014.

[AV21] Arian Askari and Suzan Verberne. Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieval. In

Proceedings of the Second International Conference on Design of Experimental Search & Information REtrieval Systems, Padova, Italy, September 15-18, 2021, volume 2950 of *CEUR Workshop Proceedings*, pages 162–170. CEUR-WS.org, 2021.

[AVA22] Amin Abolghasemi, Suzan Verberne, and Leif Azzopardi. Improving bert-based query-by-document retrieval with multi-task optimization. In *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part II*, volume 13186 of *Lecture Notes in Computer Science*, pages 3–12. Springer, 2022.

[AVA+23] Arian Askari, Suzan Verberne, Amin Abolghasemi, Wessel Kraaij, and Gabriella Pasi. Retrieval for extremely long queries and documents with RPRS: a highly efficient and effective transformer-based re-ranker. *arXiv Computing Research Repository (CoRR)*, abs/2303.01200, 2023.

[AVJ10] Leif Azzopardi, Wim Vanderbauwhede, and Hideo Joho. Search system requirements of patent analysts. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 775–776. ACM, 2010.

[AW13] Kevin D. Ashley and Vern R. Walker. From information retrieval (IR) to argument retrieval (AR) for legal cases: Report on a baseline study. In *Legal Knowledge and Information Systems - JURIX 2013: The Twenty-Sixth Annual Conference, December 11-13, 2013, University of Bologna, Italy*, volume 259 of *Frontiers in Artificial Intelligence and Applications*, pages 29–38. IOS Press, 2013.

[AYF+11] Doreen Alberts, Cynthia Barcelon Yang, Denise Fobare-DePonio, Ken Koubek, Suzanne Robins, Matthew Rodgers, Edlyn Simmons, and Dominic DeMarco. Introduction to patent searching. In *Current Challenges in Patent Information Retrieval*, volume 29 of *The Information Retrieval Series*, pages 3–43. Springer, 2011.

[AYWY+19] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. Applying BERT to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 19–24. Association for Computational Linguistics, November 2019.

[AYYZL19] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language*

Processing (EMNLP-IJCNLP), pages 3490–3496. Association for Computational Linguistics, November 2019.

[AZH+23] Sophia Althammer, Guido Zuccon, Sebastian Hofstätter, Suzan Verberne, and Allan Hanbury. Annotating data for fine-tuning a neural ranker? current active learning strategies are not better than random selection. In *Proceedings of the 1st International ACM SIGIR Conference on Information Retrieval in the Asia Pacific (SIGIR-AP'23)*, 2023.

[AZK+20] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net, 2020.

[BBCK17] Rohit Borah, Andrew W Brown, Patrice L Capers, and Kathryn A Kaiser. Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the prospero registry. *BMJ Open*, 7(2), 2017.

[BCC10] Dario Bonino, Alberto Ciaramella, and Fulvio Corno. Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics. *World Patent Information*, 32:30–38, 03 2010.

[BCS+08] Peter Bailey, Nick Craswell, Ian Soboroff, Paul Thomas, Arjen P. de Vries, and Emine Yilmaz. Relevance assessment: Are judges exchangeable and does it matter. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 667–674. Association for Computing Machinery, 2008.

[BDH03] Luiz André Barroso, Jeffrey Dean, and Urs Hölzle. Web search for a planet: The google cluster architecture. *IEEE Micro*, 23(2):22–28, 2003.

[BDSV07] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen M. Voorhees. Bias and the limits of pooling for large collections. *Information Retrieval*, 10(6):491–508, 2007.

[BGC10] Hilda Bastian, Paul Glasziou, and Iain Chalmers. Seventy-five trials and eleven systematic reviews a day: How will we ever keep up? *PLoS Medicine*, 7(9), 9 2010.

[BGG+19] Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. Fire 2019 aila track: Artificial intelligence for legal assistance. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, page 4–6. Association for Computing Machinery, 2019.

[BGPG22] Paheli Bhattacharya, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. Legal case document similarity: You need both network and text. *Information Processing Management*, 59(6):103069, 2022.

[BJ12] Peter Bailey and Li Jiang. User task understanding: a web search engine perspective. October 2012. Presentation delivered at the NII Shonan: Whole-Session Evaluation of Interactive Information Retrieval Systems workshop.

[BK08] Michael Bendersky and Oren Kurland. Utilizing passage-based language models for document retrieval. In *Advances in Information Retrieval*, pages 162–174. Springer Berlin Heidelberg, 2008.

[BLC19] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620. Association for Computational Linguistics, November 2019.

[BMR+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[Bor03] Pia Borlund. The concept of relevance in ir. *Journal of the Association for Information Science and Technology*, 54(10):913–925, August 2003.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, April 1998.

[BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv Computing Research Repository (CoRR)*, abs/2004.05150, 2020.

[BR11] Ricardo Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[Bro02] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, sep 2002.

[BT07] Jason R. Baron and Paul Thompson. The search problem posed by large heterogeneous data sets in litigation: possible future approaches to research. In *The Eleventh International Conference on Artificial Intelligence and Law, Proceedings of the Conference*, pages 141–147. Association for Computing Machinery, 2007.

[BTM+18] Patrice Bellot, Chiraz Trabelsi, Josiane Mothe, Fionn Murtagh, Jian-Yun Nie, Laure Soulier, Eric SanJuan, Linda Cappellato, and Nicola Ferro, editors. *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 9th International Conference of the CLEF Association, CLEF 2018, Avignon, France, September 10-14, 2018, Proceedings*, volume 11018 of *Lecture Notes in Computer Science*. Springer, 2018.

[Bur10] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *MSR-Tech Report*, 2010.

[BV04] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 25–32. Association for Computing Machinery, 2004.

[BWMK09] Daniela Becks, Christa Womser-Hacker, Thomas Mandl, and Ralph Kölle. Patent retrieval experiments in the context of the CLEF IP track 2009. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments, 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2, 2009, Revised Selected Papers*, volume 6241 of *Lecture Notes in Computer Science*, pages 491–496. Springer, 2009.

[Bys07] Katriina Byström. Approaches to task in contemporary information studies. *Information Research*, 12(4), 2007.

[Cas03] Donald O. Case. Looking for information—a survey of research on information seeking, needs, and behavior. *Information Ressearch*, 8:284–289, 01 2003.

[CBLL20] Hung Chau, Saeid Balaneshin, Kai Liu, and Ondrej Linda. Understanding the tradeoff between cost and quality of expert annotations for keyphrase extraction. In *Proceedings of the 14th Linguistic Annotation Workshop*, pages 74–86. Association for Computational Linguistics, December 2020.

[CBVC+18] Chris Cooper, Andrew Booth, Jo Varley-Campbell, Nicky Britten, and Ruth Garside. Defining the process to literature searching in systematic reviews: A literature review of guidance and supporting studies. *BMC Medical Research Methodology*, 18(1), 8 2018.

[CC07] Kevyn Collins-Thompson and Jamie Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in*

Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007, pages 303–310. Assocation for Computing Machinery, 2007.

[CCB09] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759. Association for Computing Machinery, 2009.

[CCHJ94] James J Cimino, Paul D Clayton, George Hripcsak, and Stephen B Johnson. Knowledge-based approaches to the maintenance of a large controlled medical terminology. *Journal of the American Medical Informatics Association*, 1(1):35–50, 1994.

[CCV12] Charles Clarke, Nick Craswell, and Ellen M. Voorhees. Overview of the TREC 2012 Web Track. In *Text Retrieval Conference (TREC)*, 2012.

[CdR16] Fei Cai and Maarten de Rijke. A survey of query auto completion in information retrieval. *Foundations and Trends in Information Retrieval*, 10(4):273–363, 2016.

[CFB+20] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282. Association for Computational Linguistics, 2020.

[CFM+20] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904. Association for Computational Linguistics, November 2020.

[CGJ96] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[CGZW11] Peng Cai, Wei Gao, Aoying Zhou, and Kam-Fai Wong. Relevant knowledge helps in choosing right teacher: Active query selection for ranking adaptation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 115–124. Association for Computing Machinery, 2011.

[CH02] Nick Craswell and David Hawking. Overview of the TREC-2002 web track. In *Proceedings of The Eleventh Text REtrieval Conference, TREC 2002, Gaithersburg, Maryland, USA, November 19-22, 2002*, volume 500-251 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2002.

[CHR01] Nick Craswell, David Hawking, and Stephen Robertson. Effective site finding using link anchor information. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 250–257. Association for Computing Machinery, 2001.

[CKP18] Jon Chamberlain, Udo Kruschwitz, and Massimo Poesio. Optimising crowdsourcing efficiency: Amplifying human computation with validation. *it - Information Technology*, 60(1):41–49, 2018.

[Cla18a] Charles L. A. Clarke. Web question answering. In *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[Cla18b] Nigel S. Clarke. The basics of patent searching. *World Patent Information*, 54:S4–S10, 2018.

[Cle91] Cyril W. Cleverdon. The significance of the cranfield tests on index languages. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '91, page 3–12. Association for Computing Machinery, 1991.

[CLvR98] Fabio Crestani, Mounia Lalmas, and Cornelis Joost van Rijsbergen. *Information retrieval: Uncertainty and logics: Uncertainty and logics: Advanced models for the representation and retrieval of information*, volume 4. Springer Science & Business Media, 1998.

[CMdR15] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. *Click Models for Web Search*. Morgan & Claypool, 2015.

[CML+21] Daniel Cohen, Bhaskar Mitra, Oleg Lesota, Navid Rekabsaz, and Carsten Eickhoff. Not all relevance scores are equal: Efficient uncertainty and calibration modeling for deep retrieval models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 654–664. Association for Computing Machinery, 2021.

[CMS19] Fabio Crestani, Stefano Mizzaro, and Ivan Scagnetto. Mobile information retrieval. *arXiv Computing Research Repository (CoRR)*, abs/1902.01790, 2019.

[CMS21] Charles L. A. Clarke, Maria Maistro, and Mark D. Smucker. Overview of the TREC 2021 health misinformation track. In *Proceedings of the Thirtieth Text REtrieval Conference, TREC 2021*, volume 500-335 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2021.

[CMY+21a] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. Ms marco: Benchmarking ranking models in the large-data regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research*

and Development in Information Retrieval, SIGIR '21, pages 1566–1576. Association for Computing Machinery, 2021.

[CMY+21b] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Ellen M. Voorhees, and Ian Soboroff. TREC Deep Learning Track: Reusable Test Collections in the Large Data Regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2369–2375. Association for Computing Machinery, 2021.

[CMYC19] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2019 deep learning track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019*, NIST Special Publication. National Institute of Standards and Technology (NIST), 2019.

[CMYC20] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2020 Deep Learning Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020*, NIST Special Publication. National Institute of Standards and Technology (NIST), 2020.

[Coh60] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[Con10] Jack G. Conrad. E-Discovery revisited: The need for artificial intelligence beyond information retrieval. *Artificial Intelligence and Law*, 18(4):321–345, 12 2010.

[CPK16] Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. Phrase detectives corpus 1.0 crowdsourced anaphoric coreference. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016*. European Language Resources Association (ELRA), 2016.

[Cra09] Nick Craswell. Mean reciprocal rank. In *Encyclopedia of Database Systems*, pages 1703–1703. Springer US, 2009.

[CRM+22] Aaron Daniel Cohen, Adam Roberts, Alejandra Molina, Alena Butryna, Alicia Jin, Apoorv Kulshreshtha, Ben Hutchinson, Ben Zevenbergen, Blaise Hilary Aguera-Arcas, Chung ching Chang, Claire Cui, Cosmo Du, Daniel De Freitas Adiwardana, Dehao Chen, Dmitry (Dima) Lepikhin, Ed H. Chi, Erin Hoffman-John, Heng-Tze Cheng, Hongrae Lee, Igor Krivokon, James Qin, Jamie Hall, Joe Fenton, Johnny Soraker, Kathy Meier-Hellstern, Kristen Olson, Lora Mois Aroyo, Maarten Paul Bosma, Marc Joseph Pickett, Marcelo Amorim Menegali, Marian Croak, Mark Díaz, Matthew Lamm, Maxim Krikun, Meredith Ringel Morris, Noam Shazeer, Quoc V. Le, Rachel Bernstein, Ravi Rajakumar, Ray Kurzweil, Romal Thoppilan, Steven Zheng, Taylor Bos, Toju Duke, Tulsee Doshi, Vincent Y. Zhao, Vinodkumar Prabhakaran, Will Rusch, YaGuang Li,

Yanping Huang, Yanqi Zhou, Yuanzhong Xu, and Zhifeng Chen. Lamda: Language models for dialog applications. *arXiv Computing Research Repository (CoRR)*, 2022.

[CvMG+14] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics, October 2014.

[CWCT23] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv Computing Research Repository (CoRR)*, abs/2306.15595, 2023.

[DC08] Pinar Donmez and Jaime G. Carbonell. Optimizing estimated loss reduction for active sampling in rank learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 248–255. Assocation for Computing Machinery, 2008.

[DC09] Pinar Donmez and Jaime G. Carbonell. Active sampling for rank learning via optimizing the area under the roc curve. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 78–89. Springer-Verlag, 2009.

[DC19] Zhuyun Dai and Jamie Callan. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 985–988. Association for Computing Machinery, 2019.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, June 2019.

[DCZ+10] Anlei Dong, Yi Chang, Zhaohui Zheng, Gilad Mishne, Jing Bai, Ruiqiang Zhang, Karolina Buchner, Ciya Liao, and Fernando Diaz. Towards recency ranking in web search. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, page 11–20. Association for Computing Machinery, 2010.

[DGGCMV+08] Manuel Carlos Díaz-Galiano, MA García-Cumbreras, María Teresa Martín-Valdivia, Arturo Montejo-Ráez, and LA Urena-López. Integrating mesh ontology to improve medical information retrieval. In *Advances in Multilingual*

and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, pages 601–606. Springer, 2008.

[dHSMM15] Alba Garcia Seco de Herrera, Roger Schaer, Dimitrios Markonis, and Henning Müller. Comparing fusion techniques for the imageclef 2013 medical case retrieval task. *Computerized Medical Imaging and Graphics*, 39:46–54, 2015.

[Dum13] Susan Dumais. Task-based search: A search engine perspective. March 2013. Invited Talk at NSF Task-Based Information Search Systems Workshop.

[DXC20] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. TREC cast 2019: The conversational assistance track overview. *arXiv Computing Research Repository (CoRR)*, abs/2003.13624, 2020.

[DZM+23] Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net, 2023.

[EDHG+20] Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962. Association for Computational Linguistics, November 2020.

[FAPH22] Maik Fröbe, Christopher Akiki, Martin Potthast, and Matthias Hagen. How train–test leakage affects zero-shot retrieval. In *String Processing and Information Retrieval*, pages 147–161. Springer International Publishing, 2022.

[FEL19] Alexander Frummet, David Elsweiler, and Bernd Ludwig. Detecting domain-specific information needs in conversational search dialogues. In *Proceedings of the 3rd Workshop on Natural Language for Artificial Intelligence co-located with the 18th International Conference of the Italian Association for Artificial Intelligence (AIIA 2019), Rende, Italy, November 19th-22nd, 2019*, volume 2521 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[FEL22] Alexander Frummet, David Elsweiler, and Bernd Ludwig. "What Can I Cook with these Ingredients?" - Understanding Cooking-Related Information Needs in Conversational Search. *ACM Transactions on Information Systems*, 40(4):81:1–81:32, 2022.

[FK01] Maria Fasli and Udo Kruschwitz. Using implicit relevance feedback in a web search assistant. In *Web Intelligence: Research and Development, First Asia-Pacific Conference, WI 2001*, volume 2198 of *Lecture Notes in Computer Science*, pages 356–360. Springer, 2001.

[FSMZ10] Donghui Feng, James Shanahan, Nate Murray, and Rémi Zajac. Learning a query parser for local web search. In *Proceedings of the 4th IEEE International Conference on Semantic Computing (ICSC 2010)*, pages 420–423. IEEE Computer Society, 2010.

[FSST97] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[GAvR09] Erik Graf, Leif Azzopardi, and Keith van Rijsbergen. Automatically generating queries for prior art search. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments, 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009*, volume 6241 of *Lecture Notes in Computer Science*, pages 480–490. Springer, 2009.

[GC11] Maura R Grossman and Gordon V Cormack. Technology-assisted review in e-discovery can be more effective and more efficient than exhaustive manual review. *Richmond Journal of Law and Technology*, XVII(3):1–49, 2011.

[GC21] Luyu Gao and Jamie Callan. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993. Association for Computational Linguistics, November 2021.

[GC22] Luyu Gao and Jamie Callan. Long document re-ranking with modular re-ranker. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2371–2376. Association of Computing Machinery, 2022.

[GCHO11] Maura R. Grossman, Gordon V. Cormack, Bruce Hedin, and Douglas W. Oard. Overview of the TREC 2011 legal track. In *Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011*, volume 500-296 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2011.

[GCR16] Maura R. Grossman, Gordon V. Cormack, and Adam Roegiest. TREC 2016 total recall track overview. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, volume 500-321 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2016.

[GDC20] Luyu Gao, Zhuyun Dai, and Jamie Callan. Modularized transfomer-based ranking framework. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4180–4190. Association for Computational Linguistics, 2020.

[GDFC20] Luyu Gao, Zhuyun Dai, Zhen Fan, and Jamie Callan. Complementing lexical retrieval with semantic residual embedding. *arXiv Computing Research Repository (CoRR)*, abs/2004.13969, 2020.

[GDS+23] Petra Galuscáková, Romain Deveaud, Gabriela González Sáez, Philippe Mulhem, Lorraine Goeuriot, Florina Piroi, and Martin Popel. Longeval-retrieval: French-english dynamic test collection for continuous web search evaluation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 3086–3094. Association of Computing Machinery, 2023.

[GG16] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org, 2016.

[GJK+16] Lorraine Goeuriot, Gareth J. F. Jones, Liadh Kelly, Henning Müller, and Justin Zobel. Medical information retrieval: introduction to the special issue. *Information Retrieval Journal*, 19(1-2):1–5, 2016.

[GKL14] Lorraine Goeuriot, Liadh Kelly, and Johannes Leveling. An analysis of query difficulty for information retrieval in the medical domain. In *Proceedings of the 37th International ACM SIGIR Conference on Research amp; Development in Information Retrieval*, SIGIR '14, page 1007–1010. Association for Computing Machinery, 2014.

[GL10] Catherine Grady and Matthew Lease. Crowdsourcing document relevance assessment with Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 172–179. Association for Computational Linguistics, June 2010.

[Glo23] Ondřej Glogar. The Concept of Legal Language: What Makes Legal Language 'Legal'? *International Journal for the Semiotics of Law - Revue internationale de Sémiotique juridique*, 36:1081–1107, 2023.

[GM22] Prashansa Gupta and Sean MacAvaney. On survivorship bias in ms marco. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pages 2214–2219. Association for Computing Machinery, 2022.

[GNS+19] Jiaming Gao, Hui Ning, Huilin Sun, Ruifeng Liu, Zhongyuan Han, Leilei Kong, and Haoliang Qi. Fire2019@aila: Legal retrieval based on information retrieval model. In *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation*, volume 2517 of *CEUR Workshop Proceedings*, pages 64–69. CEUR-WS.org, 2019.

[Goo] Google search statistics and facts 2023 (you must know). https://firstsiteguide.com/google-search-stats/. Accessed: 2023-03-15.

[GPTR10] Julien Gobeill, Emilie Pasche, Douglas Teodoro, and Patrick Ruch. Simple pre and post processing strategies for patent searching in clef intellectual property track 2009. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 444–451. Springer Berlin Heidelberg, 2010.

[GR12] Julien Gobeill and Patrick Ruch. Bitem site report for the claims to passage task in CLEF-IP 2012. In *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes*, volume 1178 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[GTC+21] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1), oct 2021.

[HAS+20] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving efficient neural ranking models with cross-architecture knowledge distillation. *ArXiv Computing Research Repository (CoRR)*, abs/2010.02666, 2020.

[HASH22] Sebastian Hofstätter, Sophia Althammer, Mete Sertkan, and Allan Hanbury. Establishing strong baselines for tripclick health retrieval. In *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Proceedings, Part II*, volume 13186 of *Lecture Notes in Computer Science*, pages 144–152. Springer, 2022.

[Hel23] Google Search Help. How google's featured snippets work. https://support.google.com/websearch/answer/9351707?hl=en, 2023. Accessed: 11-03-2023.

[Her20] William Hersh. *Information Retrieval: A Biomedical and Health Perspective*. Health Informatics. Springer International Publishing, 2020.

[HH19] Sebastian Hofstätter and Allan Hanbury. Let's measure run time! extending the IR replicability infrastructure to include performance aspects. In *Proceedings of the Open-Source IR Replicability Challenge co-located with 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, OSIRRC@SIGIR 2019, Paris, France, July 25, 2019*, volume 2409 of *CEUR Workshop Proceedings*, pages 12–16. CEUR-WS.org, 2019.

[HK22] Philipp Hartl and Udo Kruschwitz. Applying automatic text summarization for fake news detection. In *Proceedings of the Thirteenth Language Resources and*

Evaluation Conference, LREC 2022, pages 2702–2713. European Language Resources Association, 2022.

[HKA+22] Sebastian Hofstätter, Omar Khattab, Sophia Althammer, Mete Sertkan, and Allan Hanbury. Introducing neural bag of whole-words with colberter: Contextualized late interactions using enhanced reduction. ArXiv, 2022.

[HLY+21] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122. Association of Computing Machinery, 2021.

[HMRS14] Katja Hofmann, Bhaskar Mitra, Filip Radlinski, and Milad Shokouhi. An eye-tracking study of user interactions with query auto completion. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014*, pages 549–558. Association of Computing Machinery, 2014.

[HN99] Allan Heydon and Marc Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, 1999.

[HNR07] David Hunt, Long Nguyen, and Matthew Rodgers. *Patent Searching: Tools and Techniques*. Wiley, 1st edition, 2007.

[How95] Howard Turtle. Text Retrieval in the Legal World. *Artificial Intelligence and Law*, 3:5–54, 1995.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[HTBO09a] Bruce Hedin, Stephen Tomlinson, Jason R. Baron, and Douglas W. Oard. Overview of the TREC 2009 legal track. In *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009*, volume 500-278 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2009.

[HTBO09b] Bruce Hedin, Stephen Tomlinson, Jason R. Baron, and Douglas W. Oard. Overview of the TREC 2009 legal track. In *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, November 17-20, 2009*, volume 500-278 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2009.

[HTC+23] JPT Higgins, J Thomas, J Chandler, M Cumpston, T Li, MJ Page, and VA Welch. *Cochrane Handbook for Systematic Reviews of Interventions*. Cochrane, 6.2 edition, 2 2023.

[HVCB99] David Hawking, Ellen M. Voorhees, Nick Craswell, and Peter Bailey. Overview of the TREC-8 web track. In *Proceedings of The Eighth Text REtrieval Conference, TREC 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 1999.

[HYX+22] Xiaomeng Hu, Shi Yu, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu, and Ge Yu. P3 ranker: Mitigating the gaps between pre-training and ranking fine-tuning with prompt-based learning and pre-finetuning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pages 1956–1962. Association for Computing Machinery, 2022.

[HZH20a] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. Interpretable & time-budget-constrained contextualization for re-ranking. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 513–520. IOS Press, 2020.

[HZH20b] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. Neural-ir-explorer: A content-focused tool to explore neural re-ranking results. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Proceedings, Part II*, page 459–464. Springer-Verlag, 2020.

[HZS+20] Sebastian Hofstätter, Markus Zlabinger, Mete Sertkan, Michael Schröder, and Allan Hanbury. Fine-grained relevance annotations for multi-task document ranking and question answering. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management*, pages 3031–3038. Assocation of Computing Machinery, 2020.

[IJ05] Peter Ingwersen and Kalervo Järvelin. Information retrieval in context: Irix. *SIGIR Forum*, 39(2):31–39, 2005.

[JAKTV03] Peter Jackson, Khalid Al-Kofahi, Alex Tyrrell, and Arun Vachher. Information extraction from case law and retrieval of prior cases. In *Artificial Intelligence*, volume 150, pages 239–290, 11 2003.

[JDJ19] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[JGP+05] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, page 154–161. Association for Computing Machinery, 2005.

[JGP+17] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. *SIGIR Forum*, 51(1):4–11, 2017.

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[JK17] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. *SIGIR Forum*, 51(2):243–250, 2017.

[Joa02] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142. Association of Computing Machinery, 2002.

[JOH+19] Hyunhoon Jung, Changhoon Oh, Gilhwan Hwang, Cindy Yoonjung Oh, Joonhwan Lee, and Bongwon Suh. Tell me more: Understanding user interaction of smart speaker news powered by conversational search. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019*. Assocation of Computing Machinery, 2019.

[JRJ+09] Akers Jo, Aguiar-Ibáñez Raquel, Burch Jane, Chambers Duncan, Eastwood Alison, Fayter Debra, Susanne Hempel, Kate Light, Stephen Rice, Amber Rithalia, Lesley Stewart, Christian Stock, Paul Wilson, and Nerys Woolacott. *Systematic Reviews: CRD's guidance for undertaking reviews in health care*. CRD, University of York, 1 2009.

[KCS08] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008*, pages 453–456. Association of Computing Machinery, 2008.

[Ken38] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[KH17] Udo Kruschwitz and Charlie Hull. Searching the enterprise. *Foundations and Trends in Information Retrieval*, 11(1):1–142, 2017.

[KJ19] Alok Khode and Sagar Jambhorkar. Effect of technical domains and patent structure on patent information retrieval. *International Journal of Engineering and Advanced Technology*, 9(1):6067–6074, 2019.

[KJC+21] Jussi Karlgren, Rosie Jones, Ben Carterette, Ann Clifton, Edgar Tanaka, Maria Eskevich, Gareth J. F. Jones, and Sravana Reddy. TREC 2021 podcasts track overview. In *Proceedings of the Thirtieth Text REtrieval Conference, TREC 2021*, volume 500-335 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2021.

[KKM11] Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. Worker types and personality traits in crowdsourcing relevance labels. In *Proceedings of the 20th*

ACM Conference on Information and Knowledge Management, CIKM 2011, pages 1941–1944. Association of Computing Machinery, 2011.

[KKT09] Jaap Kamps, Marijn Koolen, and Andrew Trotman. Comparative analysis of clicks and judgments for IR evaluation. In *Proceedings of the 2009 workshop on Web Search Click Data, WSCD@WSDM 2009*, pages 80–87. Association of Computing Machinery, 2009.

[KOM+20] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. Association for Computational Linguistics, November 2020.

[KT01] C. C. Kuhlthau and S. L. Tama. Information search process of lawyers: A call for 'just for me' information services. In *Journal of Documentation*, volume 57, page 25–43, 2001.

[KT03] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum*, 37(2):18–28, sep 2003.

[Kuh91] Carol C Kuhlthau. Inside the search process: Information seeking from the user's perspective. *Journal of the American society for information science*, 42(5):361–371, 1991.

[KYD22] Makoto P. Kato, Takehiro Yamamoto, and Zhicheng Dou, editors. *Proceedings of the 16th NTCIR Conference Evaluation of Information Access Technologies*, 2022.

[KZ20] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020*, pages 39–48. Association of Computing Machinery, 2020.

[Lan16] Esther Landhuis. Scientific literature: Information overload. *Nature*, 535:457–458, 2016.

[LB94] Henry J Lowe and G Octo Barnett. Understanding and using the medical subject headings (mesh) vocabulary to perform literature searches. *Jama*, 271(14):1103–1108, 1994.

[LBC+15] Bo Long, Jiang Bian, Olivier Chapelle, Ya Zhang, Yoshiyuki Inagaki, and Yi Chang. Active learning for ranking through expected loss optimization. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1180–1191, 2015.

[LC94] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Morgan Kaufmann, 1994.

[LC02] Xiaoyong Liu and W. Bruce Croft. Passage retrieval based on language models. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, page 375–382. Association for Computing Machinery, 2002.

[LCG+20] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[Lee97] Joon Ho Lee. Analyses of multiple evidence combination. *SIGIR Forum*, 31(SI):267–276, July 1997.

[LETC21] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021.

[LG94] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 3–12. Springer-Verlag, 1994.

[LH13] Mihai Lupu and Allan Hanbury. Patent retrieval. *Foundations and Trends in Information Retrieval*, 7(1):1–97, 2013.

[LKS12] Deirdre Lungley, Udo Kruschwitz, and Dawei Song. Learning adaptive domain models from click data to bootstrap interactive web search. In *Advances in Information Retrieval - 34th European Conference on IR Research, ECIR 2012*, volume 7224 of *Lecture Notes in Computer Science*, pages 527–530. Springer, 2012.

[LKS+18] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 9748–9758, 2018.

[LLH20] Liang Liu, Lexiao Liu, and Zhongyuan Han. Query revaluation method for legal information retrieval. In *Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation*, volume 2826 of *CEUR Workshop Proceedings*, pages 18–21. CEUR-WS.org, 2020.

[LLH+23] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv Computing Research Repository (CoRR)*, abs/2307.03172, 2023.

[LMTM20] Tebo Leburu-Dingalo, Nkwebi Peace Motlogelwa, Edwin Thuma, and Monkgogi Modongo. UB at FIRE 2020 precedent and statute retrieval. In *Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation*, volume 2826 of *CEUR Workshop Proceedings*, pages 12–17. CEUR-WS.org, 2020.

[LNP+18] Claudio Lucchese, Franco Maria Nardini, Raffaele Perego, Roberto Trani, and Rossano Venturini. Efficient and effective query expansion for web search. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*, pages 1551–1554. Association of Computing Machinery, 2018.

[LNY21] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2021.

[Loc17] Zuccon Guido Scells Harrisen Locke, Daniel. Automatic query generation from legal texts for case law retrieval. *Information Retrieval Technology: 13th Asia Information Retrieval Societies Conference, AIRS 2017, Proceedings (Lecture Notes in Computer Science, Volume 10648)*, 10648:181–193, 2017.

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv Computing Research Repository (CoRR)*, abs/1907.11692, 2019.

[LPP+20] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, 2020.

[LPS96] Gloria J Leckie, Karen E Pettigrew, and Christian Sylvain. Modeling the information seeking of professionals: A general model derived from research on engineers, health care professionals, and lawyers. *The Library Quarterly*, 66(2):161–193, 1996.

[LR09] Patrice Lopez and Laurent Romary. Multiple retrieval models and regression models for prior art search. In *Working Notes for CLEF 2009 Workshop co-located with the 13th European Conference on Digital Libraries (ECDL 2009)*, volume 1175 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[LR10] Patrice Lopez and Laurent Romary. Experiments with citation mining and key-term extraction for prior art search. In *CLEF 2010 LABs and Workshops, Notebook Papers*, volume 1176 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[LRA23] Jurek Leonhardt, Koustav Rudra, and Avishek Anand. Extractive explanations for interpretable text ranking. *ACM Transactions on Information Systems*, 41(4):88:1–88:31, 2023.

[LRC+21] Oleg Lesota, Navid Rekabsaz, Daniel Cohen, Klaus Antonius Grasserbauer, Carsten Eickhoff, and Markus Schedl. A modern perspective on query likelihood with deep generative retrieval models. In *ICTIR '21: The 2021 ACM SIGIR International Conference on the Theory of Information Retrieval*, pages 185–195. Association of Computing Machinery, 2021.

[LSH14] Mihai Lupu, Michail Salampasis, and Allan Hanbury. Domain specific search. In *Professional Search in the Modern World - COST Action IC1002 on Multilingual and Multifaceted Interactive Information Access*, volume 8830 of *Lecture Notes in Computer Science*, pages 96–117. Springer, 2014.

[LYM+20] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. PARADE: passage representation aggregation for document reranking. *arXiv Computing Research Repository (CoRR)*, abs/2008.09093, 2020.

[LZ18] Daniel Locke and Guido Zuccon. A test collection for evaluating legal case law search. In *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*, pages 1261–1264. Association for Computing Machinery, Inc, 6 2018.

[MA02] Mark Montague and Javed A. Aslam. Condorcet fusion for improved retrieval. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, page 538–548. Association for Computing Machinery, 2002.

[MBB+07] Karen L. Myers, Pauline Berry, Jim Blythe, Ken Conley, Melinda T. Gervasio, Deborah L. McGuinness, David N. Morley, Avi Pfeffer, Martha E. Pollack, and Milind Tambe. An intelligent personal assistant for task and time management. *AI Magazine*, 28(2):47–61, 2007.

[MBB21] Iurii Mokrii, Leonid Boytsov, and Pavel Braslavski. A systematic evaluation of transfer learning and pseudo-labeling with bert-based ranking models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 2081–2085. Association for Computing Machinery, 2021.

[MBC+19] Chris Madge, Richard A. Bartle, Jon Chamberlain, Udo Kruschwitz, and Massimo Poesio. The design of A clicker game for text labelling. In *IEEE Conference on Games, CoG 2019*, pages 1–4. IEEE, 2019.

[MBPR07] Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. Automatic detection of arguments in legal texts. In *The Eleventh International Conference on Artificial Intelligence and Law, Proceedings of the Conference*, pages 225–230. Association of Computing Machinery, 2007.

[MC02] Elena Marecková and L Cervenỳ. Latin as the language of medical terminology: some remarks on its role and prospects. *Swiss medical weekly*, 132(4142):581–581, 2002.

[MGHC13] Parvaz Mahdabi, Shima Gerani, Jimmy Xiangji Huang, and Fabio Crestani. Leveraging conceptual lexicon: Query disambiguation using proximity information for patent retrieval. *SIGIR 2013 - Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122, 2013.

[MJ10] Walid Magdy and Gareth J. F. Jones. Examining the robustness of evaluation metrics for patent retrieval with incomplete relevance judgements. In *Multilingual and Multimodal Information Access Evaluation, International Conference of the Cross-Language Evaluation Forum, CLEF 2010*, volume 6360 of *Lecture Notes in Computer Science*, pages 82–93. Springer, 2010.

[MLJ09] Walid Magdy, Johannes Leveling, and Gareth Jones. DCU at CLEF-IP 2009: exploring standard IR techniques on patent retrieval. In *CLEF 2009 working notes, CEUR workshop proceedings (CEUR-WS.org)*, 2009.

[MMM15] André Mourão, Flávio Martins, and João Magalhães. Multimodal medical information retrieval with unsupervised rank fusion. *Computerized Medical Imaging and Graphics*, 39:35–45, 2015. Medical visual information analysis and retrieval.

[MOMZ21] Sheshera Mysore, Tim O'Gorman, Andrew McCallum, and Hamed Zamani. Csfcube - A test collection of computer science research articles for faceted query by example. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*, 2021.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[MS08] K. Tamsin Maxwell and Burkhard Schafer. Concept and context in legal information retrieval. In *Legal Knowledge and Information Systems - JURIX 2008: The Twenty-First Annual Conference on Legal Knowledge and Information*

Systems, volume 189 of *Frontiers in Artificial Intelligence and Applications*, pages 63–72. IOS Press, 2008.

[MSL+21] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, M. Zhang, Shaoping Ma, and Yiqun Liu. Retrieving legal cases from a large-scale candidate corpus. In *Proceedings of the eighth International Competition on Legal Information Extraction/Entailment , COLIEE 2021*, 2021.

[MSW+21] Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. Lecard: A legal case retrieval dataset for chinese law system. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2342–2348. Association for Computing Machinery, 2021.

[MVBA21] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 650–663. Association for Computational Linguistics, 2021.

[MWL20] Gregor Milicic, Sina Wetzel, and Matthias Ludwig. Generic tasks for algorithms. *Future Internet*, 12(9):152, 2020.

[MY20] Yury A. Malkov and Dmitry A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020.

[MYC+19] Chris Madge, Juntao Yu, Jon Chamberlain, Udo Kruschwitz, Silviu Paun, and Massimo Poesio. Crowdsourcing and aggregating nested markable annotations. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 797–807. Association for Computational Linguistics, 2019.

[MZK+22] Seiji Maekawa, Dan Zhang, Hannah Kim, Sajjadur Rahman, and Estevam Hruschka. Low-resource interactive active labeling for fine-tuning language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3230–3242. Association for Computational Linguistics, 2022.

[Nay19] Pandu Nayak. Understanding searches better than ever before. `https://blog.google/products/search/search-language-understanding-bert/`, 2019. Accessed: 11-03-2023.

[NC19] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv Computing Research Repository (CoRR)*, abs/1901.04085, 2019.

[NFIH10] Hidetsugu Nanba, Atsushi Fujii, Makoto Iwayama, and Taiichi Hashimoto. Overview of the patent mining task at the NTCIR-8 workshop. In *Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NTCIR-8*, pages 293–302. National Institute of Informatics (NII), 2010.

[NJPL20] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718. Association for Computational Linguistics, November 2020.

[NRS+16] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[Num] How many websites are there? `https://www.statista.com/chart/19058/number-of-websites-online/`. Accessed: 2023-03-15.

[NYCL19] Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with BERT. *arXiv Computing Research Repository (CoRR)*, abs/1910.14424, 2019.

[NZG+20] Ping Nie, Yuyu Zhang, Xiubo Geng, Arun Ramamurthy, Le Song, and Daxin Jiang. DC-BERT: decoupling question and document for efficient contextual encoding. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020*, pages 1829–1832. ACM, 2020.

[OBH+10] Douglas W. Oard, Jason R. Baron, Bruce Hedin, David D. Lewis, and Stephen Tomlinson. Evaluation of information retrieval for E-discovery. *Artificial Intelligence and Law*, 18(4):347–386, 12 2010.

[Ope23] OpenAI. GPT-4 Technical Report. Technical report, 2023.

[PBMW98] Larry Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, 1998.

[PCK+15] Massimo Poesio, Jon Chamberlain, Udo Kruschwitz, Livio Robaldo, and Luca Ducceschi. Phrase detectives: Utilizing collective intelligence for internet-scale language resource creation (extended abstract). In *Proceedings of the*

Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, pages 4202–4206. AAAI Press, 2015.

[PCP+19] Massimo Poesio, Jon Chamberlain, Silviu Paun, Juntao Yu, Alexandra Uma, and Udo Kruschwitz. A crowdsourced corpus of multiple judgments and disagreement on anaphoric interpretation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 1778–1789. Association for Computational Linguistics, 2019.

[PH19] Florina Piroi and Allan Hanbury. *Multilingual Patent Text Retrieval Evaluation: CLEF–IP*, pages 365–387. Springer International Publishing, 2019.

[Pir10] Florina Piroi. CLEF-IP 2010: Retrieval experiments in the intellectual property domain. In *CLEF 2010 LABs and Workshops, Notebook Papers*, volume 1176 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[PLH13] Florina Piroi, Mihai Lupu, and Allan Hanbury. Overview of CLEF-IP 2013 lab. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 232–249. Springer Berlin Heidelberg, 2013.

[PLHZ11] Florina Piroi, Mihai Lupu, Allan Hanbury, and Veronika Zenz. CLEF-IP 2011: Retrieval in the intellectual property domain. In *CLEF 2011 Labs and Workshop, Notebook Papers*, volume 1177 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

[PST07] Wim Peters, Maria-Teresa Sagri, and Daniela Tiscornia. The structuring of legal knowledge in LOIS. *Artificial Intelligence and Law*, 15(2):117–135, 2007.

[PZB+16] Joao Palotti, Guido Zuccon, Johannes Bernhardt, Allan Hanbury, and Lorraine Goeuriot. Assessors agreement: A case study across assessor type, payment levels, query variations and relevance dimensions. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 40–53. Springer International Publishing, 2016.

[QDL+21] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 5835–5847. Association for Computational Linguistics, 2021.

[RAB+20] Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R Hersh. TREC-COVID: rationale and structure of an information retrieval shared task

for COVID-19. *Journal of the American Medical Informatics Association*, 27(9):1431–1436, 9 2020.

[RAHK20] Julian Risch, Nicolas Alder, Christoph Hewel, and Ralf Krestel. Patentmatch: A dataset for matching patent claims & prior art. *arXiv Computing Research Repository (CoRR)*, abs/2012.13919, 2020.

[RBC+08] Filip Radlinski, Andrei Z. Broder, Peter Ciccolo, Evgeniy Gabrilovich, Vanja Josifovski, and Lance Riedel. Optimizing relevance and revenue in ad search: a query substitution approach. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008*, pages 403–410. Association of Computing Machinery, 2008.

[RDV+17] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, William R. Hersh, Steven Bedrick, Alexander J. Lazar, and Shubham Pant. Overview of the TREC 2017 precision medicine track. In *Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC*, volume 500-324 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2017.

[RDV+19] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, William R. Hersh, Steven Bedrick, Alexander J. Lazar, Shubham Pant, and Funda Meric-Bernstam. Overview of the TREC 2019 precision medicine track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC*, volume 1250 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2019.

[RDV+22] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, Steven Bedrick, and William R. Hersh. Overview of the TREC 2022 clinical trials track. In *Proceedings of the Thirty-First Text REtrieval Conference, TREC 2022*, volume 500-338 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2022.

[RDVH16] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, and William R. Hersh. Overview of the TREC 2016 clinical decision support track. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016*, volume 500-321 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2016.

[RGK21] Tony Russell-Rose, Philip Gooch, and Udo Kruschwitz. Interactive query expansion for professional search applications. *arXiv Computing Research Repository (CoRR)*, abs/2106.13528, 2021.

[RGK+22] Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. Overview and discussion of the competition on legal information extraction/entailment (COLIEE) 2021. *Rev. Socionetwork Strateg.*, 16(1):111–133, 2022.

[RIS+20] Revanth Gangi Reddy, Bhavani Iyer, Md. Arafat Sultan, Rong Zhang, Avi Sil, Vittorio Castelli, Radu Florian, and Salim Roukos. End-to-end QA on COVID-19: domain adaptation with synthetic training. *arXiv Computing Research Repository (CoRR)*, abs/2012.01414, 2020.

[RK19] Julien Rossi and Evangelos Kanoulas. Legal information retrieval with generalized language models. In *Proceedings of the 6th Competition on Legal Information Extraction/Entailment, COLIEE 2019*, 2019.

[RKG+20] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. A summary of the COLIEE 2019 competition. In *New Frontiers in Artificial Intelligence*, pages 34–49. Springer International Publishing, 2020.

[RKJ08] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008*, pages 43–52. Association of Computing Machinery, 2008.

[RL04] Daniel E. Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, page 13–19. Association for Computing Machinery, 2004.

[RLS+21] Navid Rekabsaz, Oleg Lesota, Markus Schedl, Jon Brassey, and Carsten Eickhoff. Tripclick: The log files of a large health web search engine. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2507–2513. Association of Computing Machinery, 2021.

[RM] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), pages = 441–448, publisher = Morgan Kaufmann, year = 2001,.*

[RM21] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *CHI '21: CHI Conference on Human Factors in Computing Systems*, pages 314:1–314:7. Association of Computing Machinery, 2021.

[RRCA18] Tony Russell-Rose, Jon Chamberlain, and Leif Azzopardi. Information retrieval in the workplace: A comparison of professional search practices. *Information Processing and Management*, 54:1042–1057, 11 2018.

[RRM20] Tony Russell-Rose and Andrew Macfarlane. Towards Explainability in Professional Search. In *The 3rd International Workshop on ExplainAble Recommendation and Search (EARS 2020)*, volume 5, 2020.

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[RTKJ09] Daniel M. Russell, Diane Tang, Melanie Kellar, and Robin Jeffries. Task behaviors during web search: The difficulty of assigning labels. In *42st Hawaii International International Conference on Systems Science (HICSS-42 2009), Proceedings*, pages 1–5. IEEE Computer Society, 2009.

[RZ09] Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, apr 2009.

[SAC07] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, page 623–632. Association for Computing Machinery, 2007.

[Sal17] Michail Salampasis. Federated patent search. In *Current Challenges in Patent Information Retrieval*, pages 213–240. Springer Berlin Heidelberg, 2017.

[SC99] Fei Song and W. Bruce Croft. A general language model for information retrieval. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, CIKM '99, page 316–321. Association for Computing Machinery, 1999.

[SC08] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008*, pages 1070–1079. Association for Computational Linguistics, 2008.

[SC22] Nima Sadri and Gordon V. Cormack. Continuous active learning using pretrained transformers. *arXiv Computing Research repository (CoRR)*, abs/2208.06955, 2022.

[SCR07] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*, pages 1289–1296. Curran Associates, Inc., 2007.

[SDCW19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv Computing Research Repository (CoRR)*, abs/1910.01108, 2019.

[SF94] Joseph A. Shaw and Edward A. Fox. Combination of multiple searches. In *Proceedings of The Third Text REtrieval Conference, TREC 1994*, volume 500-225 of *NIST Special Publication*, pages 105–108. National Institute of Standards and Technology (NIST), 1994.

[SGHS22] Dezhao Song, Sally Gao, Baosheng He, and Frank Schilder. On the effectiveness of pre-trained language models for legal natural language processing: An empirical study. *IEEE Access*, 10:75835–75858, 2022.

[SGV14] Rodrigo M. Silva, Marcos André Gonçalves, and Adriano Veloso. A two-stage active learning method for learning to rank. *Journal of the Association for Information Science and Technology*, 65(1):109–128, 2014.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[SHG09] György Szarvas, Benjamin Herbert, and Iryna Gurevych. Prior art search using international patent classification codes and all-claims-queries. In *Working Notes for CLEF 2009 Workshop co-located with the 13th European Conference on Digital Libraries (ECDL 2009)*, volume 1175 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[SHH20] Ian Soboroff, Shudong Huang, and Donna Harman. TREC 2020 news track overview. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020.

[SK13] Sharhida Zawani Saad and Udo Kruschwitz. Exploiting click logs for adaptive intranet navigation. In *Advances in Information Retrieval - 35th European Conference on IR Research, ECIR 2013*, volume 7814 of *Lecture Notes in Computer Science*, pages 792–795. Springer, 2013.

[SL18] Aditya Siddhant and Zachary C. Lipton. Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909. Association for Computational Linguistics, 2018.

[SLK+23] Karan Samel, Cheng Li, Weize Kong, Tao Chen, Mingyang Zhang, Shaleen Kumar Gupta, Swaraj Khadanga, Wensong Xu, Xingyu Wang, Kashyap Kolipaka, Michael Bendersky, and Marc Najork. End-to-end query term weighting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023*, pages 4778–4786. Association of Computing Machinery, 2023.

[SLM+20] Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Thuir@coliee-2020: Leveraging semantic understanding and exact matching

for legal case retrieval and entailment. *arXiv Computing Research Repository (CoRR)*, abs/2012.13102, 2020.

[SML+20] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. BERT-PLI: modeling paragraph-level interactions for legal case retrieval. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3501–3507. ijcai.org, 2020.

[SNP22] Christopher Schröder, Andreas Niekler, and Martin Potthast. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203. Association for Computational Linguistics, 2022.

[Sob17] Ian Soboroff. Building test collections: An interactive guide for students and others without their own evaluation conference series. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1407–1410. Association for Computing Machinery, 2017.

[Sob21] Ian Soboroff. Overview of TREC 2021. In *Proceedings of the Thirtieth Text REtrieval Conference, TREC 2021*, volume 500-335 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2021.

[SOJN08] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, October 2008.

[SOS92] H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992*, pages 287–294. Association of Computing Machinery, 1992.

[SPK+21] Artem Shelmanov, Dmitry Puzyrev, Lyubov Kupriyanova, Denis Belyakov, Daniil Larionov, Nikita Khromov, Olga Kozlova, Ekaterina Artemova, Dmitry V. Dylov, and Alexander Panchenko. Active learning for sequence tagging with deep pre-trained models and bayesian uncertainty estimates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021*, pages 1698–1712. Association for Computational Linguistics, 2021.

[SR17] Laura Sbaffi and Jennifer Rowley. Trust and credibility in web-based health information: A review and agenda for future research. *Journal of Medical Internet Research*, 19(6):e218, 2017.

[SS07] Mark Sanderson and Ian Soboroff. Problems with Kendall's Tau. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 839–840. Association for Computing Machinery, 2007.

[Sta21] Internet Live Stats. Total number of websites. `https://www.internetlivestats.com/total-number-of-websites/`, 2021. [Online; accessed 17-June-2021].

[Sta22] Vasileios Stamatis. End to end neural retrieval for patent prior art search. In *Advances in Information Retrieval*, pages 537–544. Springer International Publishing, 2022.

[SW21] Chirag Shah and Ryen W. White. *Task Intelligence for Search and Recommendation*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2021.

[SWJS01] Amanda Spink, Dietmar Wolfram, Major B. J. Jansen, and Tefko Saracevic. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52(3):226 – 234, 2001.

[SWY75] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[SZ05] Xuehua Shen and ChengXiang Zhai. Active Feedback in Ad Hoc Information Retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 59–66. Association for Computing Machinery, 2005.

[SZ19] Walid Shalaby and Wlodek Zadrozny. Patent retrieval: a literature review. *Knowledge and Information Systems*, 61(2):631–660, 2019.

[SZK21] Harrisen Scells, Guido Zuccon, and Bevan Koopman. A comparison of automatic boolean query formulation for systematic reviews. *Information Retrieval Journal*, 24(1):3–28, 2021.

[SZKC20] Harrisen Scells, Guido Zuccon, Bevan Koopman, and Justin Clark. Automatic boolean query formulation for systematic review literature search. In *WWW '20: The Web Conference 2020*, pages 1071–1081. Assocation of Computing Machinery / IW3C2, 2020.

[SZZ22] Harrisen Scells, Shengyao Zhuang, and Guido Zuccon. Reduce, reuse, recycle: Green information retrieval research. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2825–2837. Association of Computing Machinery, 2022.

[Tai14] John Tait. An introduction to professional search. In *Professional Search in the Modern World - COST Action IC1002 on Multilingual and Multifaceted Interactive Information Access*, volume 8830 of *Lecture Notes in Computer Science*, pages 1–5. Springer, 2014.

[TC96] Howard R. Turtle and W. Bruce Croft. Uncertainty in information retrieval systems. In *Uncertainty Management in Information Systems: From Needs to Solution*, pages 189–224. Kluwer Academic Publishers, Boston, 1996.

[Tie00] Peter Meijes Tiersma. *Legal Language*. Paperback. Chicago: The University of Chicago Press, 2000.

[TNS19] Vu Tran, Minh Le Nguyen, and Ken Satoh. Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, ICAIL '19, page 275–282. Association for Computing Machinery, 2019.

[Ton22] Nicola Tonellotto. Lecture notes on neural information retrieval. *arXiv Computing Research Repository (CoRR)*, abs/2207.13443, 2022.

[TRDG21] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. Augmented SBERT: data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 296–310. Association for Computational Linguistics, 2021.

[TRR+21] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*, 2021.

[TTSS+20] Hillel Taub-Tabib, Micah Shlain, Shoval Sadde, Dan Lahav, Matan Eyal, Yaara Cohen, and Yoav Goldberg. Interactive Extractive Search over Biomedical Corpora. In *Proceedings ofthe BioNLP 2020 workshop*, pages 28–37. Association for Computational Linguistics (ACL), 6 2020.

[Tur94] Howard Turtle. Natural Language vs. Boolean Query Evaluation: A Comparison of Retrieval Performance. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 212–220. Springer London, 1994.

[Tur95] Howard R. Turtle. Text retrieval in the legal world. *Artificial Intelligence and Law*, 3(1-2):5–54, 1995.

[ULH19] Julián Urbano, Harlley Lima, and Alan Hanjalic. Statistical significance testing in information retrieval: An empirical analysis of type i, type ii and type iii errors. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 505–514. Association for Computing Machinery, 2019.

[UPV21] Rishabh Upadhyay, Gabriella Pasi, and Marco Viviani. An overview on evaluation labs and open issues in health-related credible information retrieval. In *Proceedings of the 11th Italian Information Retrieval Workshop 2021*, volume 2947 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021.

[VB02] Ellen M. Voorhees and Chris Buckley. The effect of topic set size on retrieval experiment error. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 316–323. Association for Computing Machinery, 2002.

[VD09] Suzan Verberne and Eva D'hondt. Prior art retrieval using the claims section as a bag of words. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments, 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009*, volume 6241 of *Lecture Notes in Computer Science*, pages 497–501. Springer, 2009.

[VHW+19] Suzan Verberne, Jiyin He, Gineke Wiggers, Tony Russell-Rose, Udo Kruschwitz, and Arjen P. de Vries. Information search in a professional context - exploring a collection of professional search tasks. *arXiv Computing Research Repository (CoRR)*, abs/1905.04577, 2019.

[Voo98] Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 315–323. Association for Computing Machinery, 1998.

[Voo00] Ellen M. Voorhees. Overview of the TREC-9 question answering track. In *Proceedings of The Ninth Text REtrieval Conference, TREC*, volume 500-249 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2000.

[Voo18] Ellen M. Voorhees. On building fair and reusable test collections using bandit techniques. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 407–416. Association for Computing Machinery, 2018.

[vOS17] Marc van Opijnen and Cristiana Santos. On the concept of relevance in legal information retrieval. *Artificial Intelligence and Law*, 25(1):65–87, 3 2017.

[VP17] Marco Viviani and Gabriella Pasi. Credibility in social media: opinions, news, and health information - a survey. *WIREs Data Mining Knowl. Discov.*, 7(5), 2017.

[VR21] Ellen M. Voorhees and Kirk Roberts. On the quality of the TREC-COVID IR test collections. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2422–2428. Association of Computing Machinery, 2021.

[VSL22] Ellen M. Voorhees, Ian Soboroff, and Jimmy Lin. Can old TREC collections reliably evaluate modern neural retrieval models? *arXiv Computing Research Repository (CoRR)*, abs/2201.11086, 2022.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008, 2017.

[VvdBWK17] Suzan Verberne, Antal van den Bosch, Sander Wubben, and Emiel Krahmer. Automatic summarization of domain-specific forum threads: Collecting reference data. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, CHIIR 2017*, pages 253–256. Association of Computing Machinery, 2017.

[WA10] Metti Zakaria Wanagiri and Mirna Adriani. Prior art retrieval using various patent document fields contents. In *CLEF 2010 LABs and Workshops, Notebook Papers*, volume 1176 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[Whi13] Ryen White. Beliefs and biases in web search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, page 3–12. Association for Computing Machinery, 2013.

[Wig23] Gineke Wiggers. *The relevance of impact: biblimetric-enhanced legal information retrieval*. PhD thesis, Leiden University, 2023.

[wip04] *WIPO Intellectual Property Handbook*. WIPO publication No. 489 (E), 2004.

[WLC+20] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex Wade, Kuansan Wang, Nancy Xin Ru Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. Cord-19: The covid-19 open research dataset, 2020.

[WLZ23] Shuai Wang, Hang Li, and Guido Zuccon. Mesh suggester: A library and system for mesh term suggestion for systematic review boolean query construction. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023*, pages 1176–1179. Association of Computing Machinery, 2023.

[WML+19] Zhijing Wu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Investigating passage-level relevance and its role in document-level relevance judgment. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 605–614. Association for Computing Machinery, 2019.

[WML+20] Zhijing Wu, Jiaxin Mao, Yiqun Liu, Jingtao Zhan, Yukun Zheng, Min Zhang, and Shaoping Ma. Leveraging passage-level cumulative gain for document ranking. In *Proceedings of The Web Conference 2020*, WWW '20, page 2421–2431. Association for Computing Machinery, 2020.

[WSKZ22] Shuai Wang, Harrisen Scells, Bevan Koopman, and Guido Zuccon. Neural rankers for effective screening prioritisation in medical systematic review literature search. In *Proceedings of the 26th Australasian Document Computing Symposium, ADCS 2022*, pages 4:1–4:10. Association of Computing Machinery, 2022.

[WTRG22] Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. GPL: generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*, pages 2345–2360. Association for Computational Linguistics, 2022.

[Wu12] Shengli Wu. Ranking-based fusion. In *Data Fusion in Information Retrieval*, pages 135–147. Springer Berlin Heidelberg, 2012.

[WV20] Gineke Wiggers and Suzan Verberne. Usage and citation metrics for ranking algorithms in legal information retrieval systems. In *Proceedings of the 10th International Workshop on Bibliometric-enhanced Information Retrieval co-located with 42nd European Conference on Information Retrieval, BIR@ECIR 2020*, volume 2591 of *CEUR Workshop Proceedings*, pages 42–52. CEUR-WS.org, 2020.

[WWS+22] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[WZ09] Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on*

Research and Development in Information Retrieval, SIGIR 2009, pages 115–122. Association of Computing Machinery, 2009.

[XAZ07] Zuobing Xu, Ram Akella, and Yi Zhang. Incorporating diversity and density in active learning for relevance feedback. In *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007s*, volume 4425 of *Lecture Notes in Computer Science*, pages 246–257. Springer, 2007.

[XLS+20] Chenyan Xiong, Zhenghao Liu, Si Sun, Zhuyun Dai, Kaitao Zhang, Shi Yu, Zhiyuan Liu, Hoifung Poon, Jianfeng Gao, and Paul Bennett. CMT in TREC-COVID round 2: Mitigating the generalization gaps from web to special domain search. *arXiv Computing Research Repository (CoRR)*, abs/2011.01580, 2020.

[XSW21] Honghe Li Ning Ding Xinzhi Song, Nan Jiang and Deliang Wen. Medical professionalism research characteristics and hotspots: a 10-year bibliometric analysis of publications from 2010 to 2019. *Scientometrics*, 126(9):8009–8027, 2021.

[XXL+20] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. 02 2020.

[XXL+21] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021.

[XXS+22] Ji Xin, Chenyan Xiong, Ashwin Srinivasan, Ankita Sharma, Damien Jose, and Paul Bennett. Zero-shot dense retrieval with momentum adversarial domain invariant representations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4008–4020. Association for Computational Linguistics, May 2022.

[XYT+03] Zhao Xu, Kai Yu, Volker Tresp, Xiaowei Xu, and Jizhi Wang. Representative sampling for text classification using support vector machines. In *Advances in Information Retrieval, 25th European Conference on IR Research, ECIR 2003*, volume 2633 of *Lecture Notes in Computer Science*, pages 393–407. Springer, 2003.

[YHT+16] Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, Jean-Marc Langlois, and Yi Chang. Ranking relevance in yahoo search. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 323–332. Association for Computing Machinery, 2016.

[YKZ+22] Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. Actune: Uncertainty-based active self-training for active fine-tuning of pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*, pages 1422–1436. Association for Computational Linguistics, 2022.

[YLB20] Michelle Yuan, Hsuan-Tien Lin, and Jordan L. Boyd-Graber. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 7935–7948. Association for Computational Linguistics, 2020.

[YLF21] Eugene Yang, David D. Lewis, and Ophir Frieder. On minimizing cost in legal document review workflows. In *DocEng '21: ACM Symposium on Document Engineering 2021*, pages 30:1–30:10. Association of Computing Machinery, 2021.

[YMLF22] Eugene Yang, Sean MacAvaney, David D. Lewis, and Ophir Frieder. Goldilocks: Just-right tuning of bert for technology-assisted review. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I*, pages 502–517. Springer-Verlag, 2022.

[YPC+23] Juntao Yu, Silviu Paun, Maris Camilleri, Paloma Carretero Garcia, Jon Chamberlain, Udo Kruschwitz, and Massimo Poesio. Aggregating crowdsourced and automatic judgments to scale up a corpus of anaphoric reference for fiction and wikipedia texts. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023*, pages 767–781. Association for Computational Linguistics, 2023.

[Yu05] Hwanjo Yu. Svm selective sampling for ranking with application to data retrieval. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 354–363. Association for Computing Machinery, 2005.

[YWGH09] Linjun Yang, Li Wang, Bo Geng, and Xian-Sheng Hua. Query sampling for ranking learning in web search. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 754–755. Association for Computing Machinery, 2009.

[YXL+19] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77. Association for Computational Linguistics, June 2019.

[YZL+20] Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1725–1734. Association for Computing Machinery, 2020.

[ZA10] Guido Zuccon and Leif Azzopardi. Using the quantum probability ranking principle to rank interdependent documents. In *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010*, volume 5993 of *Lecture Notes in Computer Science*, pages 357–369. Springer, 2010.

[ZCZ+23] Yanan Zhang, Weijie Cui, Yangfan Zhang, Xiaoling Bai, Zhe Zhang, Jin Ma, Xiang Chen, and Tianhua Zhou. Event-centric query expansion in web search. In *Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023*, pages 464–475. Association for Computational Linguistics, 2023.

[ZFM+19] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343. Association for Computational Linguistics, July 2019.

[Zhd19] Fedor Zhdanov. Diverse mini-batch active learning. *arXiv Computing Research Repository (CoRR)*, abs/1901.05954, 2019.

[ZLW17] Ye Zhang, Matthew Lease, and Byron C. Wallace. Active discriminative text representation learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3386–3392. AAAI Press, 2017.

[ZML+21] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1503–1512. Association for Computing Machinery, 2021.

[ZNL+19] Zicheng Zhao, Hui Ning, Liang Liu, Chengzhe Huang, Leilei Kong, Yong Han, and Zhongyuan Han. Fire2019@aila: Legal information retrieval using improved BM25. In *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation*, volume 2517 of *CEUR Workshop Proceedings*, pages 40–45. CEUR-WS.org, 2019.

[Zob98] Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 307–314. Association for Computing Machinery, 1998.

[ZQJ+23] Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. Rankt5: Fine-tuning t5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2308–2313. Association for Computing Machinery, 2023.

[ZSH22] Zhisong Zhang, Emma Strubell, and Eduard H. Hovy. A survey of active learning for natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, pages 6166–6190. Association for Computational Linguistics, 2022.

[ZTDR23] Hamed Zamani, Johanne R. Trippas, Jeff Dalton, and Filip Radlinski. Conversational information seeking. *Foundations and Trends in Information Retrieval*, 17(3-4):244–456, 2023.

[ZWCT09] Jianhan Zhu, Jun Wang, Ingemar J. Cox, and Michael J. Taylor. Risky business: modeling and exploiting uncertainty in information retrieval. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009*, pages 99–106. Association of Computing Machinery, 2009.

[ZWYT08] Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K. Tsou. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference*, pages 1137–1144, 2008.

[ZXM+22] Jingtao Zhan, Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Evaluating interpolation and extrapolation performance of neural retrieval models. In *Proceedings of the 31st ACM International Conference on Information amp; Knowledge Management*, CIKM '22, page 2486–2496. Association for Computing Machinery, 2022.

[ZYL20] Xinyu Zhang, Andrew Yates, and Jimmy Lin. A little bit is worse than none: Ranking with limited training data. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 107–112. Association for Computational Linguistics, November 2020.

[ZYL21] Xinyu Zhang, Andrew Yates, and Jimmy Lin. Comparing score aggregation approaches for document retrieval with pretrained transformers. In *Advances in Information Retrieval*, pages 150–163. Springer International Publishing, 2021.

[ZZS+22] Munazza Zaib, Wei Emma Zhang, Quan Z. Sheng, Adnan Mahmood, and Yang Zhang. Conversational question answering: a survey. *Knowledge and Information Systems*, 64(12):3151–3195, 2022.

Appendix

Appendix A: Annotation Guidelines for TripJudge Annotation Campaign

These are the annotation guidelines, that the non-expert and expert participants of the annotation campaign saw before being directed to the annotation task:

Welcome to Fira! Our goal is to create fine-grained relevance annotations for query - document snippet pairs.

In the annotation interface you will see 1 query and 1 document snippet and a range of relevance classes to select.

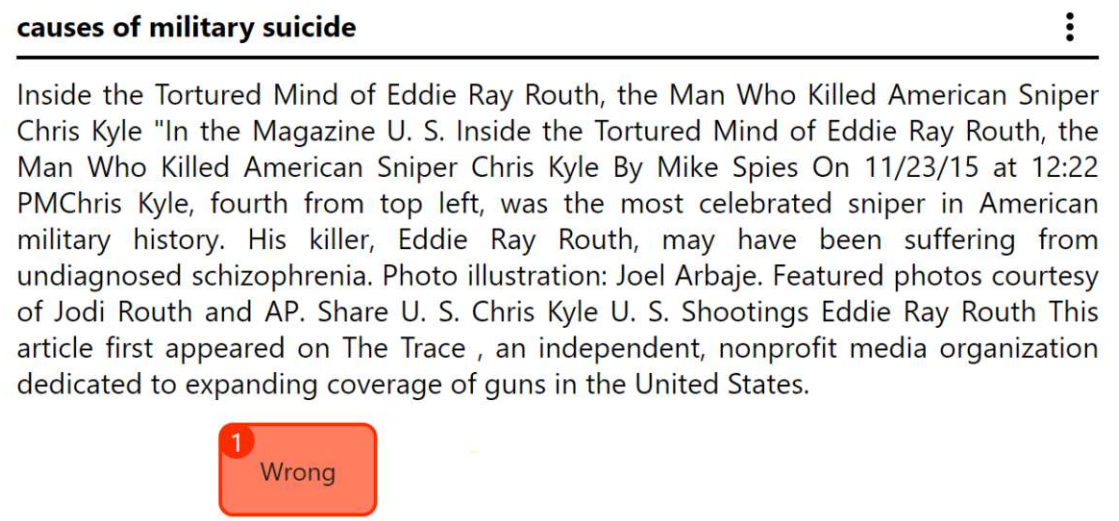
For each pair you must select 1 from 4 relevance classes: - **Wrong** If the document has nothing to do with the query, and does not help in any way to answer it - **Topic** If the document talks about the general area or topic of a query, might provide some background info, but ultimately does not answer it - **Partial** The document contains a partial answer, but you think that there should be more to it - **Perfect** The document contains a full answer: easy to understand and it directly answers the question in full

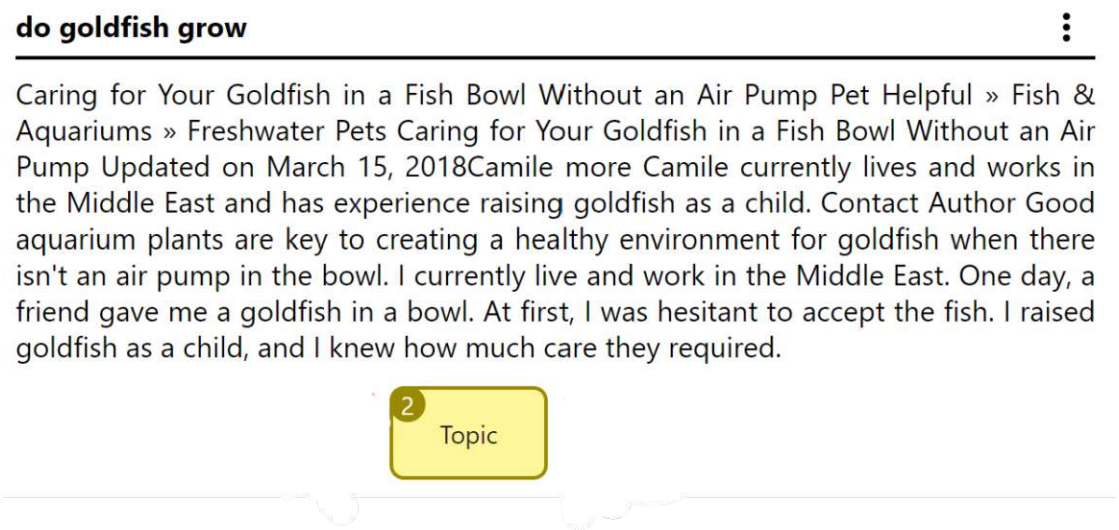
Important annotation guidelines and Fira usage tips:

(1) You should use your general knowledge to deduce links between query and answers, but if you don't know what the question (or part of it such as an acronym) means, " + 'fall back to see if the document clearly explains the question and answer and if not score it as **Wrong** or **Topic** only. We do not assume specific domain knowledge requirements.

(2) For **Partial** and **Perfect** grades you need to select the text spans, that are in fact the relevant text parts to the questions. You can select multiple words (the span) with your mouse or by once tapping or clicking on the start and once on the end of the span. You can select more than one and you can also select them before clicking on the grade button.

Now before we get started, let's have a look at an example from each relevance grade:





axon terminals or synaptic knob definition

bodies are located in the ventral horn of the spinal cord. The terminal region of the axon gives rise to very fine processes that run along skeletal muscle cells. Along these processes are specialized structures known as synapses. The particular synapse made between a spinal motor neuron and skeletal muscle cell is called the motor endplate because of its specific structure.. "Figure 4.1 (see enlarged view)Consequently, an understanding of this synapse leads to an understanding of the others. Therefore, we will first discuss the process of synaptic transmission at the skeletal neuromuscular junction. The features of the synaptic junction at the neuromuscular junction are shown in the figure at left. Skeletal muscle fibers are innervated by motor neurons whose cell

3 Partial

causes of left ventricular hypertrophy

Cardiovascular effects of hypertension Uncontrolled and prolonged elevation of BP can lead to a variety of changes in the myocardial structure, coronary vasculature, and conduction system of the heart. These changes in turn can lead to the development of left ventricular hypertrophy (LVH), coronary artery disease (CAD), various conduction system diseases, and systolic and diastolic dysfunction of the myocardium, complications that manifest clinically as angina or myocardial infarction , cardiac arrhythmias (especially atrial fibrillation), and congestive heart failure (CHF). Thus, hypertensive heart disease is a term applied generally to heart diseases, such as LVH (seen in the images below), coronary artery disease, cardiac arrhythmias, and CHF, that are caused by the direct or indirect effects of elevated BP.

4 Perfect