Informatics

# Ein eLearning Tool zur Identifizierung von Population, Intervention, Vergleich und Ergebnis für Medizinstudierende

MASTERARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Medizinische Informatik**

eingereicht von

**Florian Schweikert, BSc**
Matrikelnummer 00825224

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dr. Allan Hanbury
Mitwirkung: Linda Andersson, M.A.

Wien, 25. November 2020

Florian Schweikert    Allan Hanbury

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

Informatics

# An eLearning tool to identify population, intervention, comparison and outcome (sentiment) for medical students

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Medical Informatics**

by

**Florian Schweikert, BSc**
Registration Number 00825224

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr. Allan Hanbury
Assistance: Linda Andersson, M.A.

Vienna, 25th November, 2020

| | |
|---|---|
| Florian Schweikert | Allan Hanbury |

# Erklärung zur Verfassung der Arbeit

Florian Schweikert, BSc

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, 25. November 2020

Florian Schweikert

# Danksagung

Ich möchte meinem Betreuer Univ.Prof. Dr. Allan Hanbury für seinen Rat und sein Feedback danken. Ein besonderer Dank geht an Linda Andersson für ihre Betreuung, Geduld, ihr Feedback und ihre Motivation.

Ferner möchte ich Markus Zlabinger und Tobias Fink für all die ergiebigen Sitzungen des Gedankenaustausches zum Thema danken.

Mein Dank geht an meine Freundin Katharina, ohne die es mir wahrscheinlich nicht möglich gewesen wäre, diese Arbeit abzuschließen.

Nicht zuletzt möchte ich meinen langjährigen Studienfreunden Manuel Wiesinger und Christoph Roschger danken, die mir Feedback und Motivation gegeben hat.

# Acknowledgements

I would like to thank my advisor Univ.Prof. Dr. Allan Hanbury for his advise and feedback. A special thanks goes to Linda Andersson for her supervision, patience, feedback and motivation.

Furthermore, I want to thank Markus Zlabinger and Tobias Fink for all the fruitful sessions of exchanging ideas on the topic.

My gratitude goes to my girlfriend Katharina, without whom it probably would not have been possible for me to finish this thesis.

Last but not least, I want to thank my longtime university friends Manuel Wiesinger and Christoph Roschger for providing me with feedback and motivation.

# Kurzfassung

Die medizinische Forschung präsentiert ihre Resultate in Form von publizierten Papers in wissenschaftlichen Zeitschriften. Eine der wichtigsten Onlineplattformen zur Recherche der publizierten Papers ist PubMed, in der bereits vor dem offiziellen Erscheinen die Abstracts aller Artikel veröffentlicht werden. Die Verwendung dieser Daten in der evidenzbasierten Medizin ist jedoch immer noch eine extrem zeitaufwändige Aufgabe, da sich der Aufbau von Abstracts zwischen verschiedenen Studiendesigns, Fachrichtungen und Forschungsgruppen stark unterscheiden. Mit Stand 2019 enthielt PubMed 5,7 Millionen Artikel mit einer aktuellen Wachstumsrate von ungefähr 500.000 Artikel pro Jahr. Auch wenn moderne Suchmaschinen besser funktionieren als früher, ist es immer noch notwendig, dass der/die Forschende zumindest die Abstracts aller potenziell relevanten Artikel liest. Population, Interaction, Comparison, Outcome (PICO) stellen Grundelemente einer medizinischen Studie dar. Die Verwendung von maschinellem Lernen ist eine State-of-the-Art Methode für Labelvorhersagen in Texten. So ein Model lernt auf Basis von bereits annotierten Texten (Data Set) selbst Annotierungen durchzuführen. Ein Beispiel für so ein Data Set mit Annotierungen von PICO Elementen in medizinischen Texten ist ebmnlp, welches in dieser Arbeit mehrfach zum Vergleich herangezogen wurde.

Basis dieser Arbeit war das bereits aus dem KConnect Projekt vorhandene Data Set bestehend aus bereits PICO-Element annotierten Abstracts medizinischer Studien. Das für das KConnect Projekt entwickelte Annotierungstool wurde zu einem eLearning System erweitert, mit dessen Hilfe weitere Abstracts annotiert werden konnten (picoweb Data Set). Zur Vereinfachung für die Benutzer wurden *Intervention* und *Comparison* zu *Therapy* kombiniert. Im nächsten Schritt wurde ein neuronales Netz (SciBERT) mit dem picoweb Data Set trainiert. Dieses erzielte f1-Scores für die Erkennung von *Population* von 0.87 und für *Therapy* von 0.83. Zum Vergleich wurde das gleiche Modell mit dem ebmnlp Data Set trainiert, welches einen f1-Score von 0.81 für *Population* und 0.72 für *Therapy* erzielte.

# Abstract

Medical research presents its results in the form of published papers in scientific journals. One of the most important online platforms for researching published papers is PubMed, in which the abstracts of all articles are published even before their official publication. However, the use of this data in evidence-based medicine is still an extremely time-consuming task, as the structure of abstracts can vary between different study designs, disciplines and research groups greatly. As of 2019 PubMed contained 5.7 million articles with a current growth rate of approximately 500,000 documents per year. Even if modern search engines work better than in the past, it is still necessary, that the researcher at least reads the abstracts of all potentially relevant articles. Population, Interaction, Comparison, Outcome (PICO) are basic elements of medical studies. The use of machine learning is a state-of-the-art method for label predictions in texts. Such a model learns to perform annotations on the basis of already annotated texts (data set). An example of such a data set with annotations of PICO elements in medical texts is ebmnlp, which has been used for comparison several times in this paper.

The basis of this work was the data set already available from the KConnect project consisting of abstracts of medical studies with already annotated PICO elements. The annotation tool developed for the KConnect project was extended to an eLearning system, with which further abstracts could be annotated (picoweb data set). To simplify things for the users, *Intervention* and *Comparison* were combined to form *Therapy*. In the next step a neural network (SciBERT) was trained with the picoweb data set. This achieved f1-scores of 0.87 for the detection of *Population* and 0.83 for *Therapy*. For comparison, the same model was trained with the ebmnlp data set, which achieved an f1-score of 0.81 for *Population* and 0.72 for *Therapy*.

# Contents

# CHAPTER 1
# Introduction

## 1.1 Motivation and Problem Statement

This work contributes to improving how medical doctors work with evidence-based medicine (EBM).

"Evidence based medicine is about asking questions, finding and appraising the relevant data, and harnessing that information for everyday clinical practice." [RD95]

Rosenberg and Donald [RD95] formulated four steps in evidence-based medicine (EBM):

1. Formulate a clear clinical question from a patient's problem.
2. Search the literature for relevant clinical articles.
3. Evaluate (critically appraise) the evidence for its validity and usefulness.
4. Implement useful findings in clinical practice.

Even if this thesis focuses on the second step (literature search), the presented eLearning platform can help medical students to learn formulating a clear clinical question. As a source for literature PubMed[1] provides a large amount of medical texts, especially for clinical based trials, in human readable form. However, using this data in evidence-based medicine is still an extremely time-consuming task. In the past, about 2 hours were estimated to formulate a clinical question and search and evaluate existing literature [RD95]. Since the mid-nineties the number of available studies has risen enormously.

[1] `https://www.ncbi.nlm.nih.gov/pubmed/`

Even if modern search engines work better than in the past, it is still necessary to review many documents manually. There is still potential in making the search and the evaluation of relevant studies faster and less error-prone.

The PICO framework was developed to help formulating clinical questions. It defines the following four basic parts of a well-built clinical question [Ric+95].

***P**opulation* descripes the type of patients studied, *e.g. pregnant women* or *patients with type 2 diabetes*.
***I**ntervention* describes what intervention was done, *e.g. 5mg of cortisone* or *chemotherapy*.
***C**omparison* describes what the alternate treatment, that the study tested against, was, *e.g. placebo*.
***O**utcome* describes what the outcome of the intervention was, *e.g. reduced pain* (positive) or *no effect compared to control group* (negative)

Schardt *et al.* [Sch+07] have shown—even if statistically not significant—that using PICO to search for literature tends to enhance search result precision.

It is important for medical doctors and students to be able to extract this kind of information out of medical publications to be able to evaluate if the study results do apply to their current problem.

## 1.2 Contributions

The contributions of this work are two-folded. One contribution is the design of an elearning system that helps students and other interested parties to learn how to extract PICO data from abstracts. The annotated sentences, gathered in the process, are used to extend the training data of a Machine Learning (ML) model. The second contribution is the exploration of ways to merge these gathered annotations with another data set to improve the trained ML model.

**Research questions** Following are the research questions we answer in the result section.

- How to create an annotation platform that motivates medical students to annotate medical abstracts?
- How to evaluate the performance of the users to give them feedback on their learning progress?
- How does a state-of-the-art Machine Learning model perform on these annotations compared to the similar ebmnlp data set?

- What differences exist inside our data set (picoweb) compared to the similar ebmnlp data set?
- How does the model perform if sentences without PICO elements are included in the data?
- Is it possible to improve performance by enriching the annotations with PoS tags?
- How does the model perform when trained on a combination of both data sets?

## 1.3 Methodological approach

The focus of this work is to create an annotation platform that is easy to get started with and also uses gamification techniques to enhance long-term motivation. As the main goal of this thesis depends highly on the cooperation of the users, it is necessary to design a system that is beneficial for the users. For the Machine Learning part the goal is to achieve a reasonable prediction rate using the existing annotated data and enable the system to get better while used as an elearning tool. While choosing the used algorithms and hyperparameters, good precision is preferred over better recall. The reason for this is that false positives should be minimized, as a user reacts in a more negative way if wrong annotations are generated than if the system misses some. As the dataset only contains around 26000 annotations, computational speed is not a major concern.

### 1.3.1 Review/explore the data

The KConnect data set [Zla+18b], consisting of 1.5M PubMed abstracts, is used and extended. The picoweb data set consists of the existing KConnect annotations and the annotations gathered using the eLearning. The data is analysed to identify necessary preprocessing steps. These preprocessing steps are similar to the pipeline used in [Zla+18a]. Suitable unsupervised methods are used to evaluate the structure of the available text. Furthermore we analyse and compare the picoweb data set with the ebmnlp data set [Nye+18] and explore possibilities of combining them.

### 1.3.2 Develop an elearning platform

We present an easy-to-use web application, enabling users to learn to label PICO elements in PubMed abstracts. In contrast to common annotation platforms like brat[2] our approach focuses on usability over complexity. The annotation frontend created by Markus Zlabinger [Zla+18b] is used as base for the annotation part of the application. The frontend was extended and simplified to meet the requirements of an eLearning platform. The user can choose from several medical topics (*e.g. cardiac*, *psychiatry*, ...) to start a learning session. Each learning session consists of a PubMed abstract to annotate—one sentence at a time. The annotations of the users are stored to extend

[2]`https://brat.nlplab.org/` (visited 2020-09-13)

the data set and provide data to compute the performance score of new annotations. At the end of each session the users shown how well they performed by comparing their annotations with the existing ones (Section 3.2.2). To simplify the user experience Intervention and Comparison were combined into Therapy, as first user tests showed that it is often ambiguous what the *Intervention* is and what the *Comparison*—*e.g.* in a study comparing three treatments.
Annotation of Outcome was not implemented for two reasons. The first reason is that the existing KConnect data set does not include outcome either, so there were no annotations to base the initial model training on, to show predictions to the user. The second reason is to remove complexity, as our approach focuses on usability over complexity.

The system computes and displays basic statistics on how the user is performing. This information could also be used to weight the annotations as input for a machine learning model. Different gamification approaches are implemented and evaluated based on user feedback.

Finally the eLearning system was linked to a basic prediction service, using a model trained on the gathered annotations, allowing manual review of the prediction quality and also rating these predictions as correct or incorrect.

### 1.3.3 Train basic model

We already have access to annotated data from the KConnect[3] project. The data consists of 24770 annotations of 8902 sentences in 1415 documents. Over 11000 annotations are marked as intervention, over 7000 as population and over 6000 as comparison. The already existing annotated PubMed abstracts, created by Zlabinger *et al.* [Zla+18b], are used to train a neural network using SciBERT [BLC19]. By using a part of the data as test data, several approaches were evaluated. To get insights into the impact of including sentences without PICO labels or adding PoS tags, we trained the model with and without them and compared the results. We present a comparison of the model scores trained on the picoweb data set and the scores trained on the ebmnlp data set (Section 2.1.2, 4.3). Furthermore we merged both data sets and trained a model using this combined data set.

## 1.4 Structure of the work

In Chapter 2 we give a literature overview for the research questions we want to answer. Chapter 3 describes the methods used to answer these research questions. In Chapter 4 we describe our gathered data and present our answers to the research questions. In Chapter 5 we discuss our findings and future work.

[3]`http://www.kconnect.eu/` (visited on 2020-02-22)

# CHAPTER 2

# Background

As stated by Gobbel *et al.* [Gob+14] high costs are incurred because reviewers with sufficient expertise are needed to annotate medical texts. For that reason the availability of well annotated texts is rather limited. Besides the quality, the amount of data is very important in supervised machine learning to achieve good predictions. Therefore it is advisable to look into possible ways to expand the data set by combining it with other data sets from the same domain, depending on the task even data sets in a different language can be used [FES16]. Another approach is to get usable results with less labeled data. As stated by Bergsma [Ber10], standard supervised learning models can only guess the labels if a sequence of words did not occur in the training data. As stated by [KZW19] using unsupervised Machine Learning methods to initialize a model with pretrained parameters from a large unlabeled corpus significantly improves the model performance. The large amount of raw text available from different domains offer opportunities to use unsupervised Machine Learning methods to pretrain a model on the structure of the text.

Figure 2.1 illustrates what information unsupervised methods—like word2vec (see: 2.3.3)—are able to extract from unlabeled text. The example was created using the PubMed corpus used for picoweb. Digits for example are in a cluster on the top right side near some written numbers. On the right side there is a clear cluster of units too. Left from this cluster there is another cluster containing organs (kidney, heart, lung, . . . ). You will also find several pairs of two words with strong connections, *e.g.* contradistinctions (small/large), singular/plural (receptor/receptors), or words with similar meaning (ability/capacity). Be aware that this is just a two-dimensional representation of a 300-dimensional model, therefore not all connections between words are visible.

There are several approaches to improve model designs using unsupervised techniques. Using semi-supervised methods like unsupervised pre-training, it is possible to make use of unlabeled data to enhance the model. To explore the question why unsupervised pre-training works so well Erhan *et al.* [Erh+10] reviewed several possible explanations in

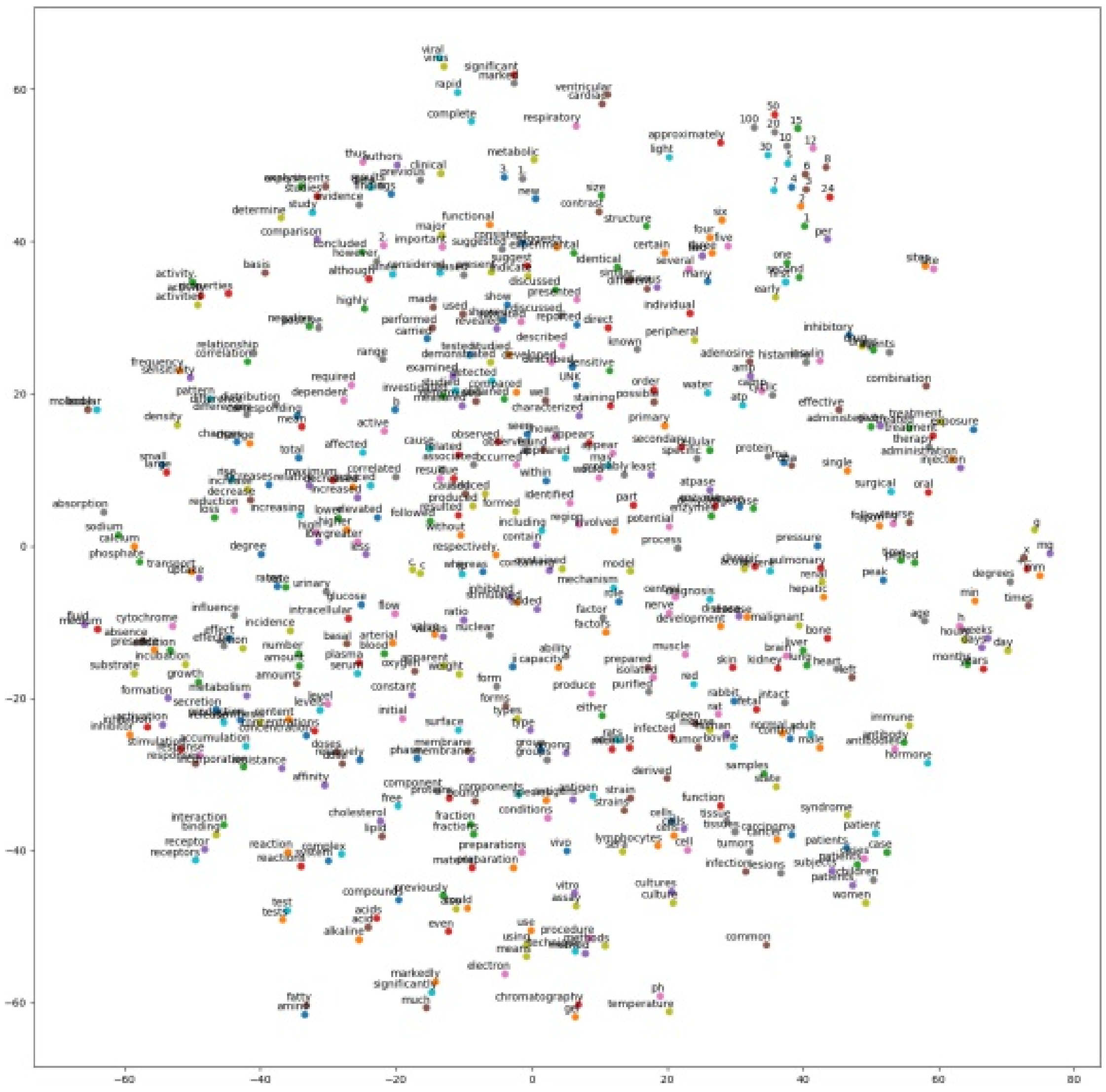


Figure 2.1: Word cluster generated from PubMed abstracts using word2vec (Section 2.3.3)

the literature and performed experiments to explore them. Their experiments were done on the MNIST digit classification and InfiniteMNIST data set. Their results support the hypothesis that, as stochastic gradient descent weight changes do have more impact in early training, it can get trapped in the basin of attraction. In a setting with a small training set, pre-training helps to find better minima in terms of generalization performance.

## 2.1 Medical texts

The structure, syntax and terminology of documents may vary according to the domains [Gob+14]. Therefore NLP systems trained on one domain may not work well for another domain [Gob+14]. In medical research, papers are often using the traditional Introduction, Methods, Results, Discussion (IMRD) format, but as the analysis of Nwogu [Nwo97] shown these elements can be further subdivided into eleven "moves", A move is a text segment that forms a unit. It is functionally related to the text of which it is a part, where not every move always exists.
Figure 2.2 shows the subdivisions Nwogu [Nwo97] describes. This results in a similar structure through different papers.

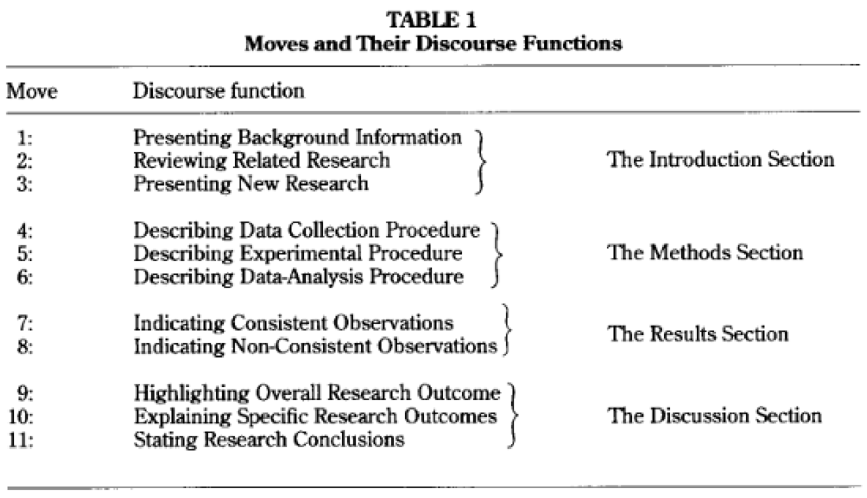


**TABLE 1**
**Moves and Their Discourse Functions**

| Move | Discourse function | |
|---|---|---|
| 1: | Presenting Background Information | The Introduction Section |
| 2: | Reviewing Related Research | |
| 3: | Presenting New Research | |
| 4: | Describing Data Collection Procedure | The Methods Section |
| 5: | Describing Experimental Procedure | |
| 6: | Describing Data-Analysis Procedure | |
| 7: | Indicating Consistent Observations | The Results Section |
| 8: | Indicating Non-Consistent Observations | |
| 9: | Highlighting Overall Research Outcome | The Discussion Section |
| 10: | Explaining Specific Research Outcomes | |
| 11: | Stating Research Conclusions | |

Figure 2.2: Moves and Their Discourse Functions ([Nwo97] page 7)

As the medical sector is subdivided in many diverse fields (*e.g.* surgery, psychiatry, . . . ) and the work of medical professions vary, so do the (finer) structure and grammar used in papers [Pet82].

### 2.1.1 Picoweb/KConnect data set

We base our work on the data gathered in a former KConnect[1] project [Zla+18b].

Zlabinger *et al.* [Zla+18b] presented their process of establishing a corpus for Population, Interaction, Comparison, Outcome to be used by medical doctors for more advanced search queries. Based on 1.5M PubMed titles and abstracts from Trip[2] they selected a subset containing only Randomized Control Trials (RCTs). An annotation platform was designed to gather the annotations from 6 persons which came from different backgrounds (linguists, biologists, medical experts and students). The interface was

[1]http://www.kconnect.eu/
[2]https://www.tripdatabase.com/

improved between three test rounds. They aimed to increase the agreement between the annotators after each interface improvement. They achieved this goal by increasing the agreement from 20% (1st version) to 55% (final version).

The resulting data set consists of 24770 annotations of 8902 sentences in 1415 documents. All documents are abstracts of Randomized Control Trials (RCTs). Over 11000 annotations are marked as intervention, over 7000 as population and over 6000 as comparison.

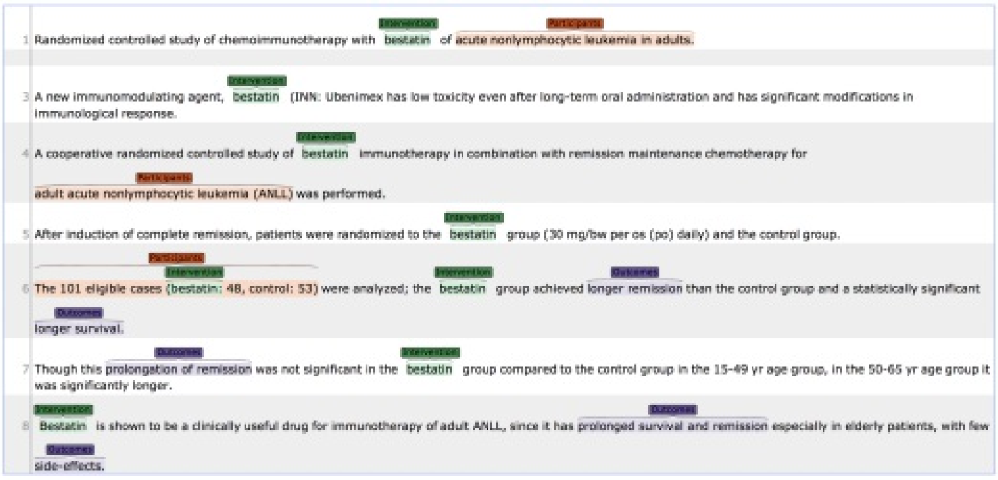
### 2.1.2 EBMNLP data set

Nye, *et al.* presented a corpus of 5000 PICO annotated abstracts of RCTs [Nye+18]. The data set consists of span based PICO annotations and more complex hierarchical labels. As we only used the span based labels, we do not describe the hierarchical data here. The span based labels are published in the CoNLL-2003 format [SD03], where Part of Speech (PoS) and chunk are not used. The Named Entity Recognition element was used for the PICO label. They used Amazon Mechanical Turk (AMT) to gather annotations of the abstracts from both non-experts (three workers for each of the 5000 abstracts) and experts (for 200 documents). Overall, the annotations were done by 579 AMT non-expert workers and two experts (medical students). As the annotations generated by the AMT workers are very noisy, they tested three different aggregation strategies to improve the data quality. These three strategies were a simple majority vote, the Dawid-Skene model [DS79], and HMM-Crowd [Ngu+17]. They choose HMM-Crowd as it delivered the best results. They outlined multiple tasks and created baseline models for each of them. The for us interesting task is identifying PICO spans in abstracts—as this is also one of the aims of this thesis. They presented two baseline models, a linear Conditional Random Field (CRF) and a Long Short-Term Memory (LSTM) - CRF model. While the CRF performed rather badly with f1-scores of 0.53, 0.32, and 0.29 (population, intervention, outcome), the LSTM model performed much better with scores of 0.71, 0.65, and 0.63.

Figure 2.3 shows an example of the span based annotation process used to create the ebmnlp data set.

### 2.1.3 Data set analysis

An often used method in Natural Language Processing (NLP) to calculate the similarity between corpora is the cosine similarity. Li and Han [LH13] have shown that even if cosine similarity has some shortcomings it is still a quite effective way to measure the similarity between corpora. For convenience we used the classic cosine similarity to analyse the data set differences. In Equation 2.1 $A$ and $B$ are two vectors with elements to be compared, where $A_i$ and $B_i$ are components of vector $A$ and $B$ respectively.



1 Randomized controlled study of chemoimmunotherapy with bestatin of acute nonlymphocytic leukemia in adults.

3 A new immunomodulating agent, bestatin (INN: Ubenimex has low toxicity even after long-term oral administration and has significant modifications in immunological response.

4 A cooperative randomized controlled study of bestatin immunotherapy in combination with remission maintenance chemotherapy for adult acute nonlymphocytic leukemia (ANLL) was performed.

5 After induction of complete remission, patients were randomized to the bestatin group (30 mg/bw per os (po) daily) and the control group.

6 The 101 eligible cases (bestatin: 48, control: 53) were analyzed; the bestatin group achieved longer remission than the control group and a statistically significant longer survival.

7 Though this prolongation of remission was not significant in the bestatin group compared to the control group in the 15-49 yr age group, in the 50-65 yr age group it was significantly longer.

8 Bestatin is shown to be a clinically useful drug for immunotherapy of adult ANLL, since it has prolonged survival and remission especially in elderly patients, with few side-effects.

Figure 2.3: ebmnlp span annotation example (`https://ebm-nlp.herokuapp.com/annotations` [visited 2020-09-13])

$$\cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}} \tag{2.1}$$

Entropy is a measure for the average level of information inside a message 2.2. In Equation 2.2 $Z$ is a vector containing events and $z$ is a single even in $Z$. $p_z$ is the probability the event $z$ happening. $b$ is the base of the logarithm. For tossing a coin $b$ would be 2 and $Z$ would contain the events that the coin is heads and the event that the coin is tails. If the coin is fair, the probability for each event is 0.5. This leads to $-(0.5\cdot\log_2 0.5 - 0.5\cdot\log_2 0.5) = 1.0$ which represents maximum uncertainty for predicting the next toss. In case the probability $p_z$ of an element is zero, the summand is set to 0: $\lim_{p\to 0^+} p\log(p) = 0$.

$$H_1 = E[I] = -\sum_{z\in Z} p_z \log_b p_z \tag{2.2}$$

## 2.2 Annotation gathering and eLearning

Even if the main target group of our approach are medical students, to enlarge the potential user group, the system should be suitable for GPs and interested persons without medical background too. As our target groups are adults the system was developed with the way adults learn (in contrast to children) in mind [Mer01]. Interactive eLearning approaches are mainly seen as an extension of classical lecturing, with the benefit of helping to maintain the interest of the learner [RML06]. We want to answer the question how to make an annotation system more appealing for the user. Also we want to explore ways to provide these annotators with enough benefits to stay motivated.

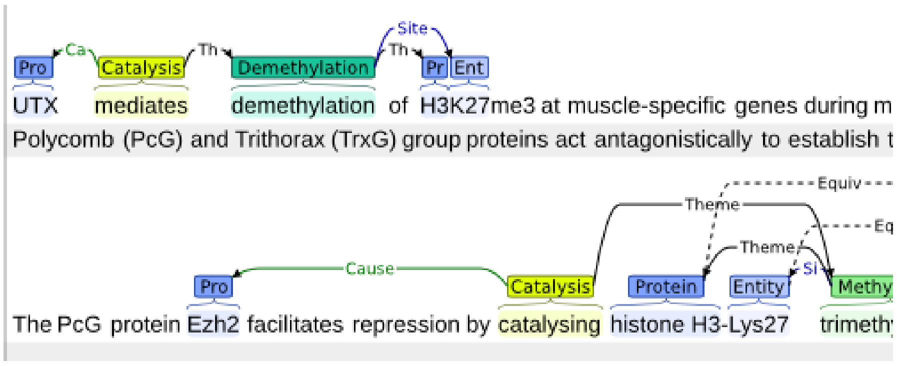

Pro
Ca
Catalysis
Th
Demethylation
Site
Th
Pr
Ent
UTX mediates demethylation of H3K27me3 at muscle-specific genes during m
Polycomb (PcG) and Trithorax (TrxG) group proteins act antagonistically to establish t
Equiv
Theme
Eq
Theme
Cause
Si
Pro
Catalysis
Protein
Entity
Methy
The PcG protein Ezh2 facilitates repression by catalysing histone H3-Lys27 trimeth

Figure 2.4: BioNLP annotation interface example using brat (`https://brat.nlplab.org/examples.html` [visited 2020-09-13])

A common tool to annotate text corpora is brat[3]. While it can be used even for complex annotation tasks, there is an initial barrier for new users. As seen in the example setup for a BioNLP task shown in Figure 2.4, the interface provides the possibility to annotate complex relations between token spans. Being able to handle this complexity requires training before a user can start annotating text.

Our approach on the other hand aims to create a system that is intuitive enough to allow a user to start annotating with as little introduction as possible. On one hand this means a reduced complexity of the gathered annotations. On the other hand this can increase the overall amount of training data, as the user needs less time to finish an annotation, and therefore can do more annotations in the same time. Of course there are use cases—like extracting deeper knowledge from texts—where the resulting data set might be insufficient, but for our targeted use case of helping MDs to find relevant papers for a specific medical question no complex relations between token spans are needed.

### 2.2.1 User performance and error rate

Measurement of how often a user is able to label a sentence correct is essential to provide the user with feedback of the learning progress. It can also be used to weight the impact future annotations of the user should have in the model training process. To answer the question of how to formalize the performance of a new annotation of an user, we decided to calculate the inter-annotator agreement (IAA) between the new annotation to the existing annotations of the same sentence. This of course only works if there already are annotations of the same sentence.

[3]`https://brat.nlplab.org/`

As we want to compare the annotations of more than two raters, Cohen's Kappa [Coh60] is not sufficient, as it is only designed for two raters. Therefore we use Fleiss' Kappa [Fle71]—which is based on the Scott's Pi [Sco55]—to calculate the IAA.

$$
\begin{aligned}
\kappa &= \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \\
p_j &= \frac{1}{Nn} \sum_{i=1}^{N} n_{ij}, \qquad 1 = \sum_{j=1}^{k} p_j \\
P_i &= \frac{1}{n(n-1)} \sum_{j=1}^{k} n_{ij}(n_{ij} - 1) \\
\bar{P} &= \frac{1}{N} \sum_{i=1}^{N} P_i, \quad \bar{P}_e = \sum_{j=1}^{k} p_j^2
\end{aligned}
\tag{2.3}
$$

Where $N$ is the total number of subjects, $n$ is the number of ratings per subject, and $k$ is the number of possible categories. $\bar{P}$ is the percentage of judgments on which the raters agree (accuracy), and $\bar{P}_e$ is the probability of an agreement by chance. $p_j$ represents the proportion of all assignments to the $j$-th category, and $P_i$ represents the extent to which annotators agree on the $i$-th subject.

### 2.2.2 Gamification

As stated in Section 1.2 we want to find a way to create an annotation platform that motivates medical students to annotate PICO elements in medical abstracts. Gamification is a popular method to create long term motivation by adding elements known from games to the platform. Deterding *et al.* [Det+11] proposed a definition of *gamification* as *the use of game design elements in non-game contexts.*

*Gamification (gameful design)* is distinct from the related concepts *(Serious) games*, *Toys*, and *Playful design*, as it uses gaming elements only as a tool to achieve a separate goal. One has to differentiate between *gaming* and *playing*, where *gaming* contains rules and is towards a discrete outcome [Cai01] (Figure 2.5).

Such gamification elements can be *e.g.* progress bars, experience points or virtual goods [Dic+15].

Dicheva *et al.* [Dic+15] reviewed 34 papers and identified 15 design principles (Table 2.1).

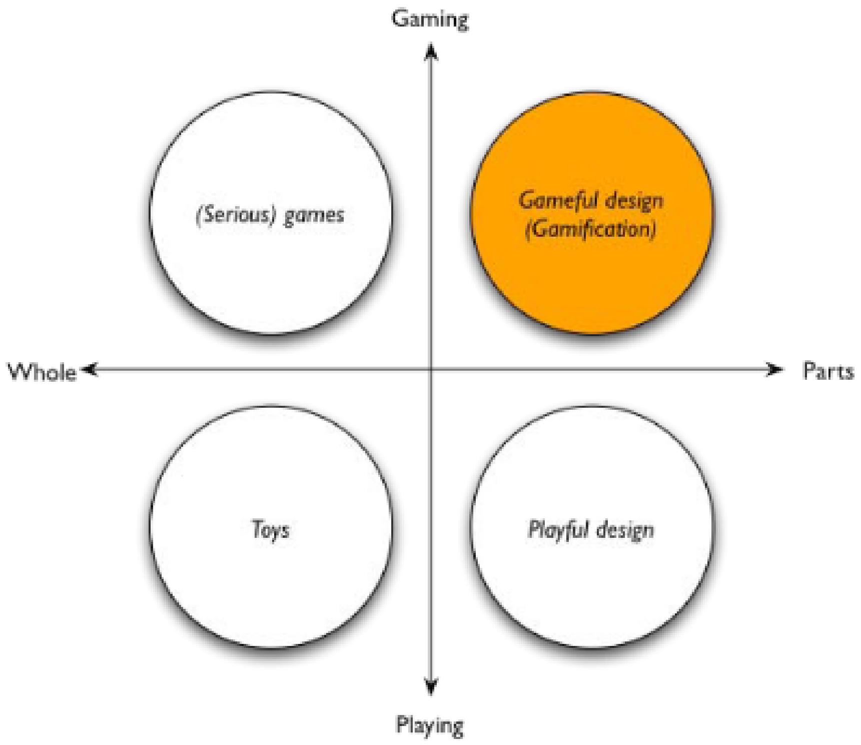


Figure 2.5: "Gamification" between game and play, whole and parts ([Det+11] page 5)

| | |
|---|---|
| Goals | mean the gamified system has clear goals the user seeks to achieve |
| Challenges and quests | mean there are self-contained tasks to do, to finish a quest |
| Customization | means the tasks are personalized to each user (*e.g.* increasing difficulty) |
| Progress | means the users get visible feedback on their progress (*e.g.* progress bars, levels) |
| Feedback | means the users get timely feedback on their performance |
| Competition and cooperation | adds a social component to the system (*e.g.* leaderboards, groups/guilds, . . . ) |
| Accrual grading | allows earning *e.g.* points for finishing tasks. |
| Visible status | is another social element, it allows the user to display their achievements and earn social reputation |
| Unlocking content | allows the user to unlock new challenges or similar by finishing tasks |
| Freedom of choice | means the user is not restricted to a linear path through the tasks |
| Freedom to fail | means failing a task does not have severe consequence, the user has *e.g.* the possibility to retry a task |
| Storytelling | embed the tasks into a bigger story framework |
| New identities | *e.g.* avatars |
| Onboarding | design principles lower the barrier of entrance for a new user (*e.g.* a tutorial for new users) |
| Time restriction | adds an additional challenge for the user by enforcing the tasks to be finished in a specific time |

Table 2.1: 15 design principles identified by Dicheva *et al.* [Dic+15]

## 2.3 Language representation models

As first step to prepare a corpus to be used by a Machine Learning algorithm, the documents have to be split into sentences and sentences into words (tokens) using a tokenizer. A tokenizer chops a sequence of characters into pieces (*tokens*) and typically removes characters like punctuation. Tokenization is more complex than simply splitting a string at each space character. It has to handle special characters like punctuation correctly (*e.g. 0.99* is a single token). An extreme example where splitting at spaces does not work is chinese. Chinese does not use spaces between characters and a word can consist of two characters, where each character also represent another word if treated as a single character word. To run learning algorithms on text data it is necessary to convert the words to some mathematical constructs like vectors. There are multiple approaches to achieve this.

### 2.3.1 One-Hot

The simplest type to represent text data in a computable form is the one-hot encoding. Each word in the text corpus gets a unique, sequential identifier (ID) assigned to it. This ID is used to create a one dimensional vector of the size of the vocabulary. Every position in the vector is 0 except for the position with the index of the word, which is 1.

This method is very easy to implement but the vectors created by it are sparse. This has the disadvantage to be unable to work with words not present in the training corpus [Seo+16]. Also One-Hot encodings do not handle any information from the word context. As our PICO extraction model has to be able to deal with new words (*e.g.* a new drug) this method is insufficient.

| Word | ID |
|---|---|
| lorem | 0 |
| ipsum | 1 |
| dolor | 2 |
| sit | 3 |
| amet | 4 |

$$\Longrightarrow \begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix} \begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix} \begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix} \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}$$

Table 2.2: One Hot Embedding

### 2.3.2 Global Vector

Global Vector (GloVe) [PSM14] is a count-based method to map words into a dense vector space. The idea is to count how often a word appears in the context of other words. Therefore a co-occurrence matrix is built for the complete corpus. Afterwards dimension reduction is used to generate a dense vector. Even if this method is a much better approach than simple one-hot, there are better methods available for our use case.

### 2.3.3 Word2Vec

Another technique for computing continuous vector representations of words is word2vec.

Mikolov *et al.* [Mik+13] propose two model architectures for computing continuous vector representations of words. They measured the quality of the representations in a word similarity test, comparing it with the previously best performing techniques. They aimed to introduce a technique that makes it possible to train on over a billion words with high dimensionality. They also wanted to make it possible to map multiple degrees of similarity [MYZ13] in this vector space. This means that not only similar used words are close together (*e.g.* countries), it also means versions of a word with different endings are close together in one dimension, and different words with the same ending are also close together in another dimension.

The authors were surprised that similarity of word representations go beyond simple syntactic regularities. So is it possible to perform algebraic operations on the vectors representing the words, *e.g.* "$Paris - France + Italy = Rome$".

The two new model architectures are called continuous bag of words (CBOW) and continuous skipgram.

**The continuous bag of words** is trained to maximize the probability to get a word as output, by its context words. This is done for each word in the corpus.

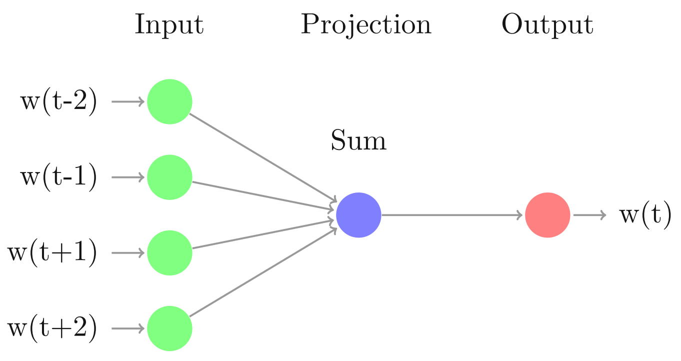


Figure 2.6: continuous bag of words (CBOW) takes two tokens before and two after a word (context) and trains to predict the word itself

**The continuous skipgram architecture** takes the opposite approach, it iterates over the words in a corpus and takes one word after another as input, to maximize the probability to get the context words as output for each input word. While this method is more complex, it improves the quality of the resulting word vectors [TSA15].

To measure the quality of the word vectors, the authors defined 869 semantic and 10675 syntactic questions. A question is answered correctly only if the closest representation of a word in the vector is the correct one. Using a 300 dimensional skip-gram with 783M

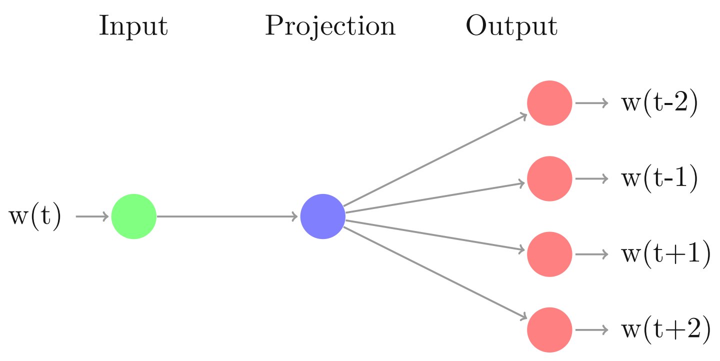


Figure 2.7: Continuous Skipgram takes a word and trains to predict two tokens before and two after a word (context)

training words, they achieved an accuracy of 50.0% for the semantic tests, compared to 34.2% using a 100 dimensional Feedforward Neural Net Language Model (NNLM) with 6B training words. For the syntactic test the skip-gram is slightly worse than the NNLM with 53.1% vs. 64.5%.

### 2.3.4 BERT

Even as word embeddings like word2vec represent a major breakthrough in Natural Language Processing and made finding word analogies possible, they have some shortcomings. One property of word embeddings like word2vec is that they treat every appearance of a word the same way, independent of the meaning in the context it is used. For example word embeddings cannot distinguish between the meanings of the word "running", as in "running a marathon" or "running a company". To solve this problem Bidirectional Encoder Representations from Transformers (BERT) was developed [Dev+18]. BERT provides a pre-trained language model trained on the BooksCorpus (800M words) [Zhu+15] and and English Wikipedia (2,500M words).

#### SciBERT

SciBERT [BLC19] is based on BERT, with the difference that it is trained using text of scientific papers. The corpus consists of 18% computer science papers and 82% papers from the biomedical domain.

### 2.3.5 Rare/unknown words problem

Classic language representation models use vocabularies to convert between words and vector space. That leads to two problems. First, context-based models cannot learn much about the typical context of a word if it is very rare. Second, not all words are part of the training data, so it is necessary to handle unknown words for doing predictions. One naive way would be to just ignore them by removing them before inserting the sentence into the model. As this would change the whole sentence structure, this is not advisable. A common way to deal with such words is to replace them with an *UNK* token. This leaves the sentence structure intact, but of course some information is lost. There are approaches to improve on just treating all *UNK* tokens the same way. As stated by Gulcehre *et al.* [Gul+16], covering a large number of words is key to building a robust NLP system. Increasing the vocabulary is problematic as it only leads to more rare words with an inaccurate vector representation. A vector representation is inaccurate if it is not placed near vector representations of similar words. Especially, if the words you want to predict are often unknown/rare words—as in [Bor+15]—this can be critical as there is not much training data to learn a accurate vector representation of a word. Gulcehre *et al.* [Gul+16] subdivided existing approaches solving the rare/unknown word problem into three categories. The approaches in the first category focus on enhancing the computation speed, allowing it to be run with a larger vocabulary. This helps mitigate the unknown word problem, but does not help for the rare word problem. The second group uses information from the word context. This works well for machine translation and question answering, by copying words from the input text. Approaches in the third category change the input/output resolution, like using the input corpus on character level. These suffer less from the rare/unknown word problem, but, because of the increased amount of input elements, training usually becomes harder.

BERT uses an approach of the third category to deal with the unknown word problem by splitting unknown words into subtokens.

## 2.4 Neural Networks & Deep Learning

The basic ideas for neural networks date back to the middle of the last century [Ros58] or even further. The availability of large amounts of data and fast processing units (especially GPUs) made it possible to readopt these ideas. In contrast to earlier Machine Learning methods deep learning does not involve extensive manual feature extraction. Instead most features are learned by finding patterns in training data [LBH15]. Reducing manual feature extraction also leads to a more unified neural network architecture usable for more than a single task, as the model also discovers patterns in the data that are probably not important for one task but for a different task [Col+11].

Deep learning (and Machine Learning in general) made major breakthroughs in many areas of computer science possible. In 2012, a deep convolutional neural network was used to win the ImageNet Large Scale Visual Recognition Competition [KSH12]. After that deep learning got more and more popular for image recognition.
Besides computer vision applications, Natural Language Processing (NLP) is an area in which deep learning offers a high potential [HM15].

Machine Learning uses features to classify elements—like a word is most likely a drug name if it is after a number and the word *mg*. Traditional Machine Learning often includes a high amount of manual feature engineering. This means that it was necessary to define rules by hand. Such manually defined rules can get very complex. Using neural

networks the trend goes to end-to-end learning. The idea behind end-to-end learning is to apply gradient-based learning to a system as a whole [Gla17]. This also includes “peripheral” parts of the model, like representation learning [Gla17].
For example, the traditional approach for speech recognition looks like this:
*Audio input* → *feature extraction* → *phoneme detection* → *word composition* → *text output*
With end-to-end learning such a system would look like this:
*Audio input* → *neural network* → *text output*
A benefit of such an end-to-end approach is to avoid task-specific engineering [Col+11].

The basic idea of (artificial) neural networks is based on how the human brain works, even if most of modern neural networks are very simplified. They consist of multiple layers of neurons connected to each other. Each neuron calculates a weighting sum of its inputs (plus a bias). An activation function then produces the output of the neuron.

Basically a neural network is an arbitrary function approximator. It is able to approximate any continuous function, even if the neural network only has one hidden layer [Cyb89].

### 2.4.1 Activation function

To achieve an approximation of a nonlinear function, the network itself needs to be nonlinear. In neural networks the activation function is the nonlinear part (*e.g.* sigmoid or ReLU).

### 2.4.2 Training

After the layers of a neural network are defined, its weights need to be initialized. This is normally done by setting the weights to small random values. Backpropagation is the dominant method to calculate the gradient of the loss function with respect to the weights. The term is also often wrongly used for the whole learning process. The backpropagation algorithm was published in 1986 [RHW86]. First the network is used feed-forward to calculate an output, based on the current weights. The output is compared to the training data and used to calculate the error using backpropagation. The weights must now be optimized to reduce the loss or in other words to match the desired output better. This is similar to classic mathematical optimization, with the difference that in Machine Learning the goal is to optimize against unseen samples instead of the training data (Section 2.4.3). To achieve this an optimizer is used. The optimizer determines how weights are adapted while training to minimize the loss or cost. A popular method to find a minima in Machine Learning is gradient descent (GD). The basic principal behind gradient descent methods is as follows. The initialized network is used to make a prediction—even if guess would be a better expression at this stage. The loss function is used to get the loss—a measure of how bad the prediction is, based on the prediction in contrast to the training data. As the gradient of the loss points in the direction of the highest increase of the loss, using the inverse points in the direction of the the highest decrease. Adapting the weights in that direction brings us in direction of a local minimum. Repeating this for each training item called an epoch. Common practice is to calculate the performance (Section 2.5) after each epoch to decide if another epoch should be trained. Even if the validation performance does not improve or even slightly decreases after an epoch it is possible it will improve again after a later epoch. Therefore it is recommended to store the model state after each epoch and keep training the model until it does not improve for multiple epochs. After the training is stopped the state of the epoch with the best validation performance is chosen.

As described by Ruder [Rud16], one challenge of gradient descent is choosing the learning rate—representing the impact of each training sample. A low learning rate obviously leads to slow convergence. On the other side if the learning rate is too high it can prevent convergence and cause the loss function to fluctuate around a local minimum. Using learning rate schedules, the learning rate is adjusted while training (*e.g.* reduced after each epoch). But as these adjustments have to be specified in advance it is not possible for this method to adapt for the characteristics of the data set. Furthermore the same learning rate is applied to all parameter updates.

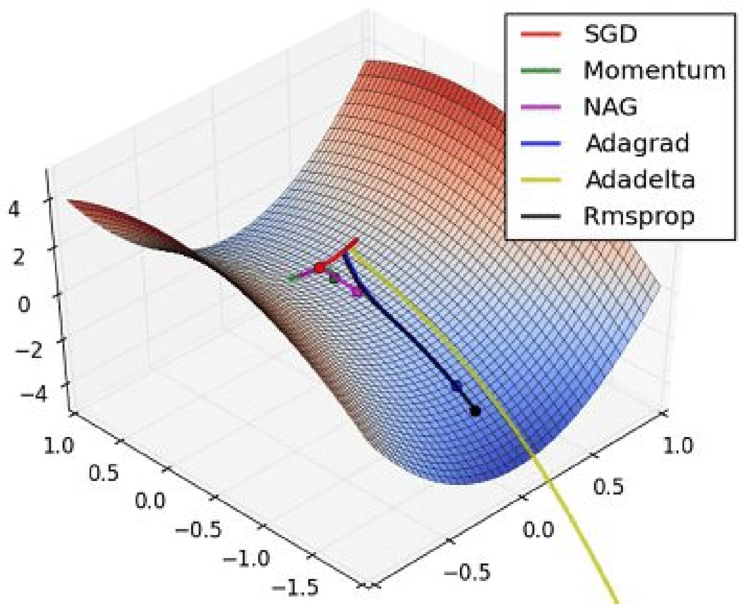


Figure 2.8: Gradient descent optimization methods ([Rud16] page 10)

Ruder presented several of the most commonly used optimizations of the learning process. We summarize the algorithms important for this thesis. Many build up on the momentum method [Qia99]. This method aims to accelerate gradient descent in the relevant directions while slowing down less relevant directions. Therefore a fraction $\gamma$ (usually about 0.9) of the update of the past step is taken into account at each step. Such a momentum method behaves similar to a ball getting faster in direction of the steepest slope (see also Figure 2.8).

$$\beta_1, \beta_2 \in [0, 1) \quad \text{exponential decay rates for the moment estimates} \tag{2.4}$$

$$m_0 = 0, v_0 = 0 \tag{2.5}$$

$$g_t = \nabla_\Theta f_t(\Theta_{t-1}) \quad \text{gradients w.r.t. stochastic objective at timestep}t \tag{2.6}$$

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad \text{biased first moment estimate} \tag{2.7}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad \text{biased second raw moment estimate} \tag{2.8}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad \text{bias-corrected first moment estimate} \tag{2.9}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad \text{bias-corrected second raw moment estimate} \tag{2.10}$$

$$\Theta_t = \Theta_{t-1} - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad \text{updated parameter} \tag{2.11}$$

Adaptive Moment Estimation (Adam) [KB14] is a method to compute adaptive learning rates for each parameter. It stores an exponential moving average of the gradient (first moment estimate $m_t$) the squared gradient (second raw moment estimate $v_t$).

Equations 2.4-2.11 describe an update step of Adam, where $\alpha$ is the step size. $m_0$ and $v_0$ are initialized as 0 (Equation 2.5), which has the effect that it leads to moment estimates that are biased towards zero. This bias are counteracted resulting in $\hat{m}_t$ (Equation 2.9) and $\hat{v}_t$ (Equation 2.10). Popular implementations of Adam often add L2 regularization by adding $\delta\Theta_{t-1}$ to the cost function to calculate $g_t$ (Equation 2.6): $g_t = \nabla_\Theta f_t(\Theta_{t-1}) + \delta\Theta_{t-1}$ [LH17].

In their presentation of Adam [KB14] Kingma and Ba suggest using $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ as good default settings. They also tested Adam with multiple popular Machine Learning models, where it performed better than AdaGrad, AdaDelta, SGDNestrov, and RMSProp.

Loshchilov and Hutter [LH17] demonstrated that adaptive gradient algorithms like Adam, do not generalize as well as SGD with momentum. They presented an adaption of Adam called *AdamW* where they decouple weight decay and loss-based gradient updates. This is achieved by moving the regularization term $\delta\Theta_{t-1}$ to parameter update $\Theta_t$ (Equation 2.11): $\Theta_t = \Theta_{t-1} - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon} + \delta\Theta_{t-1}$.

As stated by Léon Bottou in [Bot10] the computational complexity of learning algorithms became the limiting factor when dealing with large data sets. Stochastic gradient descent (SGD)—a widely used optimizer— does not perform well on large data sets. On the other hand adaptive gradient algorithms that perform better on large data sets, were shown to generalize worse than SGD with momentum. As (Sci)BERT is trained on a large corpus, while having the need for good generalization, neither classic SGD nor

Adam are a good fit. *AdamW* represents a version of Adam with better generalization capabilities, while retaining the scaling benefits.

### 2.4.3 Overfitting

Overfitting can be a problem when working with deep neural networks as it means the network is learning the noise in the training data that does not exist in the test data [Sri+14]. A model has to be able to work with data unknown at the time of training, which requires it to represent a generalization of the data. If the model is not able to generalize from the training set well, it will perform badly with new data. This happens if the model uses too many variables to describe the data and is trained for too long. In general deep neural networks tend to learn simple patterns first as the experiments in [Arp+17] show, but begin to memorize the training examples later.

> “Deep neural networks contain multiple non-linear hidden layers and this makes them very expressive models that can learn very complicated relationships between their inputs and outputs. With limited training data, however, many of these complicated relationships will be the result of sampling noise, so they will exist in the training set but not in real test data even if it is drawn from the same distribution.” [Sri+14]

One way to deal with overfitting is to use a method called dropout. With this regularization method, randomly chosen nodes are deleted in a training step [Sri+14]. In a way it is a similar approach as adding noise while training an unsupervised model.

### 2.4.4 Vanishing gradient problem

In Machine Learning using backpropagation each weight of the neural network receives an update based on the activation function with respect to the current weight. If using an activation function like the sigmoid function that use gradients between 0 and 1, multiplying multiple such gradients while propagating the weight changes back in direction to the input layer of the network, will result in a very small value when reaching the first layer. This results in very slow training of the first layers. In the worst case the Machine Learning model stops learning on new data [Hoc91][Hoc98]. As the first layers are crucial to detect core features of the input data, insufficient training of them lead to an overall bad performance of the network.

The vanishing gradient problem is often circumvented by using another activation function like ReLU. Glorot *et al.* [GBB11] presented the use of a rectified linear unit (ReLU) activation function that works better for training neural networks than sigmoid or hyperbolic tangent. They based their research on neuroscience findings. As neurons are rarely maximally saturated, it is possible to approximate it by a rectifier. They performed several experiments in image recognition and sentiment analysis. Their results show that using ReLU increases performance of image recognition and sentiment analysis tasks. This is especially true if no unsupervised pre-training were done. Using ReLU as the activation function also prevents the vanishing gradient problem and is cheaper to compute.

### 2.4.5 Bidirectional Long Short-Term Memory

When working with text it is necessary to include the context of a token into the training process, as the token on its own can be ambiguous. For example the word *present* can be used as noun with different meanings (*present* time, giving you a *present*), adjective (be *present* at an event), or verb (to *present* something). The most obvious way is to use a sliding window over the text. This implies selecting a window size, but that leads to the problem on how to find the optimal window size. Using a too small window size can make the network incapable of learning some important information from the context. Using a too large window size leads to overfitting on the training data.

A better way is to use a recurrent neural network (RNN), which is a powerful sequence model. A problem with basic RNNs is that—as it processes the input in temporal order—it only learns from past context. Another problem is—although theoretically possible—learning long range input/output dependencies using a RNN is difficult [BSF94]. Also—as they are very deep neural networks—basic RNNs suffer from the vanishing gradient problem (see: Section 2.4.4).

Graves and Schmidhuber [GS05] presented a bidirectional LSTM that solves these problems. It also outperforms unidirectional networks, and is faster and more accurate than standard RNNs.

The problem of not learning from the future context is solved by using a bidirectional RNN. The idea of a bidirectional recurrent neural network was introduced by Schuster and Paliwal [SP97] to create a model that is able to learn from future context too. The idea is to add a second RNN connected to the same output layer, but feeding it the input in reverse order. This way the network has the complete context for every point in a given sequence. They tested the new approach by running phoneme classification on the TIMIT data set[4]. Using the bidirectional RNN they achieved over 77% recall for the 61 symbol data set. Using a unidirectional approach it was less than 72%.

To solve the problem of learning long range input/output dependencies they use a Long Short-Term Memory (LSTM) model. Hochreiter and Schmidhuber introduced the LSTM in 1997 [HS97]. They introduced a novel, gradient-based method that has a computational complexity of $O(1)$ per time step and weight. LSTMs implemented an approach to achieve a constant error flow, this solves the vanishing gradient problem (see: Section 2.4.4). Simplified, an LSTM extends the recurring part of a recurrent neural network by memory cells and gates. These gates control how the memory is used. They decide when to ignore new information, forget information stored in the memory and when to select it to impact the prediction.

[4]`https://catalog.ldc.upenn.edu/LDC93S1`

## 2.5 Metrics

In this section we describe the most relevant metrics used for validating Machine Learning (ML) models.

| | **relevant** | **non-relevant** | |
|---|---|---|---|
| **retrieved** | $A \cap B$ | $A \cap \overline{B}$ | $B$ |
| **not retrieved** | $\overline{A} \cap B$ | $\overline{A} \cap \overline{B}$ | $\overline{B}$ |
| | $A$ | $\overline{A}$ | $N$ |

Table 2.3: metric sets overview ($|N|$ = number of elements in the system, $|A|$ = number of relevant elements, $|B|$ = number of retrieved elements) [JR71]

**Precision** shown in Figure 2.12 represents the ratio of how many of the retrieved elements $|B|$ are relevant ($|A \cap B|$) [JR71].

$$precision = \frac{|A \cap B|}{|B|} \tag{2.12}$$

**Recall** shown in Figure 2.13 represents the ratio of how many of the relevant elements $|A \cap B|$ are retrieved in relation to the the overall relevant elements $|A|$ [JR71].

$$recall = \frac{|A \cap B|}{|A|} \tag{2.13}$$

**F1-score** shown in Figure 2.14 is defined as a harmonic mean of precision and recall [SF07] (see: 2.14). Hand and Christen argued in [HC18] that the f1-score method has a major conceptual weakness—the relative importance of precision and recall should be an aspect of the problem. The more generic $F_\beta$ allows applying weights to value precision or recall more than the other. We did not use $F_\beta$ as the pytorch version, used by SciBERT, did not support it at this time.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2.14}$$

## 2.6 Summary

In this chapter we described the importance of unlabeled texts to learn initialization parameters for Machine Learning models. Furthermore, we presented the basic structure of medical texts as well as both the KConnect/picoweb and the ebmnlp data set. Additionally we described algorithms to analyse presented data sets and furthermore shed some light on background to gamification principles and error rate calculation used to create our eLearning system. At last we presented the background for our Machine Learning model.

CHAPTER 3

# Method

As mentioned in chapter 2 annotating medical data can be expensive, but there are many unannotated text corpora freely available. This available data is used to pre-train a model. As SciBERT provides us with a suitable pre-trained model we do not need to do pre-training ourselves (Section 3.3).

## 3.1 Data set analysis

Before we are able to analyse the corpora we need to do some basic preprocessing on the data. The documents have to be split into sentences and sentences into words (tokens) using a tokenizer. The CoreNLP toolkit [Man+14] was used as such a tokenizer.

The preprocessing was kept minimalistic and was restricted to converting the text to lowercase. Additionally the sentences are shuffled between documents before being used to train the word2vec. Otherwise, as the learning rate decreases, sentences from early documents would have more impact than sentences of later documents. This leads up to around 25% better results (UMN-SRS word similarity [Pak+10]) than using the sentences unshuffled [Chi+16]. 75% of the picoweb annotations are used as training set.

Cosine similarity was calculated using the formula described in Section 2.1.3 to get a quick overview of the similarity between tokens labeled *population* and *therapy* inside a data set. This was calculated for the picoweb and ebmnlp data set. First we created a vocabulary of all labeled words inside a data set. Afterwards we counted the times each token in the vocabulary appears labeled as *population* ($|A|$) and the times it appears as *therapy* ($|B|$). At last we put $|A|$ and $|B|$ in Equation 3.1 to calculate the cosine similarity. A high similarity means that, that the tokens in the data set appear similar often as *population* and *therapy*.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}} \tag{3.1}$$

To find out how much information a token provides inside a data set, we used the Shannon entropy [Sha48]. This was done for the picoweb and the ebmnlp data set. In Equation 3.2, $Z$ contains the probabilities of a token being *population*, *therapy*, and appearing unlabeled. As we have three probabilities we use 3 as base for the logarithm. This way a lower entropy (uncertainty) means the token is a good indicator for a label.

$$H_1 = E[I] = \sum_{z \in Z} p_z I(z) = -\sum_{z \in Z} p_z \log_3 p_z \tag{3.2}$$

## 3.2 Annotation and eLearning

Expert annotators are expensive and while using cheap non-expert annotators *e.g.* Amazon Mechanical Turk works for easy tasks, as shown by the authors of [Sno+08], there is a lot of domain-knowledge necessary for annotating medical texts [KZW19].

Medical students could be a good source of this type of annotation. They already have some medical knowledge, but are cheaper than medical experts. In fact, presenting the annotation platform as an eLearning tool, it can provide the necessary benefit for the user in non-monetary form. This section will deal with the question of how to create an annotation platform that motivates medical students to annotate medical abstracts.

### 3.2.1 Gamification

Based on the design principles described in Section 2.2.2, we chosen the most suitable principles for our use case. Table 3.1 describes which design principles are implemented for our eLearning system and which are not, including a description how they are implemented respectively why they are not implemented.

| | **Design principle** | **Description** |
|---|---|---|
| ☐ | Goals | besides self-contained annotation sessions and categories, our system does not provide any bigger goals |
| ☒ | Challenges and quests | we implemented self-contained categories as quests (Section 3.2.1), where the user gets a badge for annotating all documents in a category |
| ☐ | Customization | this was not implemented, as personalization of tasks is non-trivial and increases the complexity of the system |
| ☒ | Progress | the user can see his/her progress for each category in form of a progress bar |
| ☒ | Feedback | our system provides immediate feedback to the user after each annotation session (Section 3.2.2) |
| ☒ | Competition and cooperation | the social components in our system are limited to an all-time and a weekly leaderboard, showing the top-annotators by annotation count |
| ☐ | Accrual grading | the system counts each annotation done by a user, but currently it is only visible in the leaderboard as there is no user profile page |
| ☐ | Visible status | besides the leaderboard there is no social component where the status of the user is presented to other users |
| ☒ | Unlocking content | our system allows users to unlock new categories when finishing a category |
| ☒ | Freedom of choice | the user has the possibility to choose in which category he/she wants to annotate documents |
| ☒ | Freedom to fail | even if it is not possible to retry a task, the only negative effect of a wrong annotation is a reduced score, which can be undone by new correct annotations |
| ☐ | Storytelling | we did not implement a bigger story framework, as embedding the annotation of medical texts into such a framework is difficult |
| ☐ | New identities | this was not implemented, as the implemented social elements kept at a minimum to prevent the scope of the application getting to big |
| ☒ | Onboarding | to help the user to get started, we implemented a step-by-step tutorial, explaining how to use the system |
| ☐ | Time restriction | this was not implemented, as enforcing the user to finish an annotation faster, would enhance the chance of faulty annotations |

Table 3.1: Description of implemented design principles identified by Dicheva *et al.* [Dic+15]

### Categories

The writing style of papers vary—besides differences between authors or work groups—according to the subdomain of the work. There are no metadata like in which medical domain the study was done—also interdisciplinary studies could make it ambiguous. We analysed the widely available tags provided for PubMed abstracts.

**MeSH** The Medical Subject Headings (MeSH) tags provide a suitable way to create categories with documents of similar topics. First we experimented with an adapted k-means algorithm named k-modes[Hua98] to cluster the categorical data. The kmodes[1] library was used. The algorithm was used to split the 2550 documents of the KConnect data set into different categories. We did several clustering runs with different number of target clusters—starting from only five clusters up to twenty in steps of five. These cluster counts were chosen to get a manageable amount of categories, with around 100 to 500 abstracts each, for the user to choose from. The result showed us that the combination of MeSH tags, assigned to the categories by the kmodes library, produced highly unbalanced amounts of documents per category.

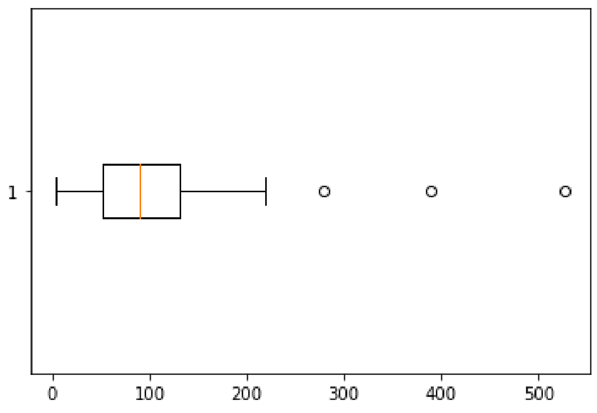


Figure 3.1: Distribution of number of cluster documents (20 categories)

This approach generated some fairly good clusters, *e.g.* for pregnancy related abstracts, where 91 of the 102 documents in the cluster had pregnancy related MeSH tags. As expected many of the clusters included similar documents, but it was hard to find good labels (category names) for the clusters, as multiple clusters shared similar "themes". The size of the clusters ranged from five documents to over 500 documents (Figure 3.1). As the automatic clustering did not show good outcome, the categories were created by assigning MeSH tags manually to the categories. Three categories were created for the test rounds using the admin interface of our eLearning system.

**Cardiology** was selected by assigning most (except *heart burn*) tags that contain *heart* or *cardiac*. We chose this category as it consists of 108 documents, which is realistic for a user to annotate. Furthermore it was easy to find an expressive category image.

Following MeSH tags were assigned for this category:

[1]`https://github.com/nicodv/kmodes/` (visited on 2019-11-02)

- (D006333) Heart Failure
- (D002303) Cardiac Output, Low
- (D006348) Cardiac Surgical Procedures
- (D002304) Cardiac Pacing, Artificial
- (D005117) Cardiac Complexes, Premature
- (D001145) Arrhythmias, Cardiac
- (D006331) Heart Diseases
- (D006352) Heart Ventricles
- (D002301) Cardiac Glycosides
- (D006328) Cardiac Catheterization
- (D002302) Cardiac Output
- (D006350) Heart Valve Prosthesis
- (D006325) Heart Atria
- (D016027) Heart Transplantation
- (D006324) Heart Arrest, Induced
- (D006330) Heart Defects, Congenital
- (D006349) Heart Valve Diseases
- (D002305) Cardiac Tamponade
- (D019918) Heart Valve Prosthesis Implantation
- (D004489) Edema, Cardiac
- (D058406) Cardiac Resynchronization Therapy

**Psychiatry** was chosen by assigning 12 psychiatry and/or psychology related MeSH tags. It contains 98 documents. We chose this category as psychiatry is a relatively young, independent field, with little risk of overlapping documents with the *Cardiology* or *Colonscopy* category.

Following MeSH tags were assigned for this category:

- (D011569) Psychiatric Status Rating Scales
- (D011613) Psychotherapy
- (D011581) Psychological Tests
- (D013315) Stress, Psychological
- (D011618) Psychotic Disorders
- (D011619) Psychotropic Drugs
- (D012565) Schizophrenic Psychology
- (D011567) Psychiatric Department, Hospital
- (D011602) Psychophysiologic Disorders
- (D003858) Dependency (Psychology)
- (D011570) Psychiatry
- (D064889) Psychotherapy, Psychodynamic

**Colonscopy** was chosen by assigning the MeSH tag *(D003113) Colonoscopy*. This category only contains 10 documents and was used to test the automatic unlocking of a new category if a user completes a category.

**Quality of Life** consists of 115 documents with the *(D011788) Quality of Life* tag assinged. This category was used to test if it gets unlocked when finishing all annotations in the *colonscopy* category.

### 3.2.2 Measuring the reliability/performance of a user

It is critical to know how well a user performs on the annotation tasks, for providing feedback to the user about his/her progress—this will answer the question on how to evaluate a users' performance and to potentially weight the annotation while training the neural network. Measuring the reliability is done by comparing the users' input with the annotations of the same sentence done by other users. Getting more annotations of the same sentence helps to reduce the impact of outliers. We use Fleiss' Kappa (Section 2.2.1) to calculate the inter-annotator agreement (IAA) between a new annotation and the existing ones for the same sentence. If less than three other users annotated the sentence we do not calculate the IAA to prevent a single wrong annotation has too much impact on the score.

### 3.2.3 Prototypes and user feedback

Based on the annotation tool of [Zla+18a] we created two prototypes of the eLearning platform.

A first version was created and tested by several users. Afterwards we held a discussion round to allow the test users to give us feedback about their experiences while using the platform. The users were asked to describe how easy it was for them to start annotating abstracts and if they had any problems while doing so. They also were asked if they have any improvement suggestions. Based on this feedback we created an improved version of the system. After the second test round we again held a a discussion round

#### First version

Three students tested the first version of the platform. Their feedback was used to improve the system for a second test run.

A new user is provided with a simple "get started" guide to learn the basic usage of the eLearning tool. This guide is always available and easily found in the main menu. Additionally the full annotation guidelines are available to download at the guide page.

To begin with each user gets three hand picked categories to work on (Section 3.2.1). The chosen categories were:

- Cardiology
- Colonoscopy
- Psychiatry

Colonoscopy was selected to provide a category with a small set of documents to test progress related features faster.

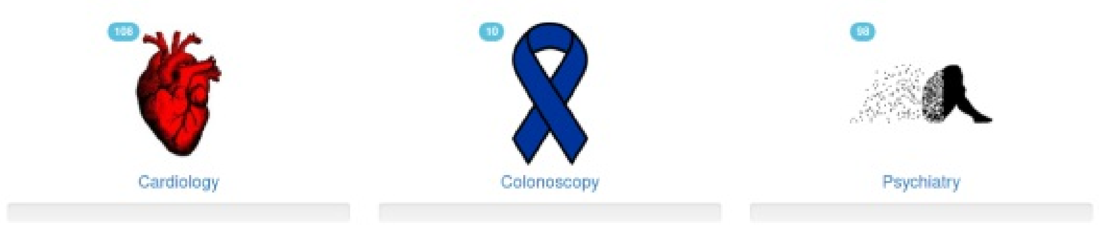
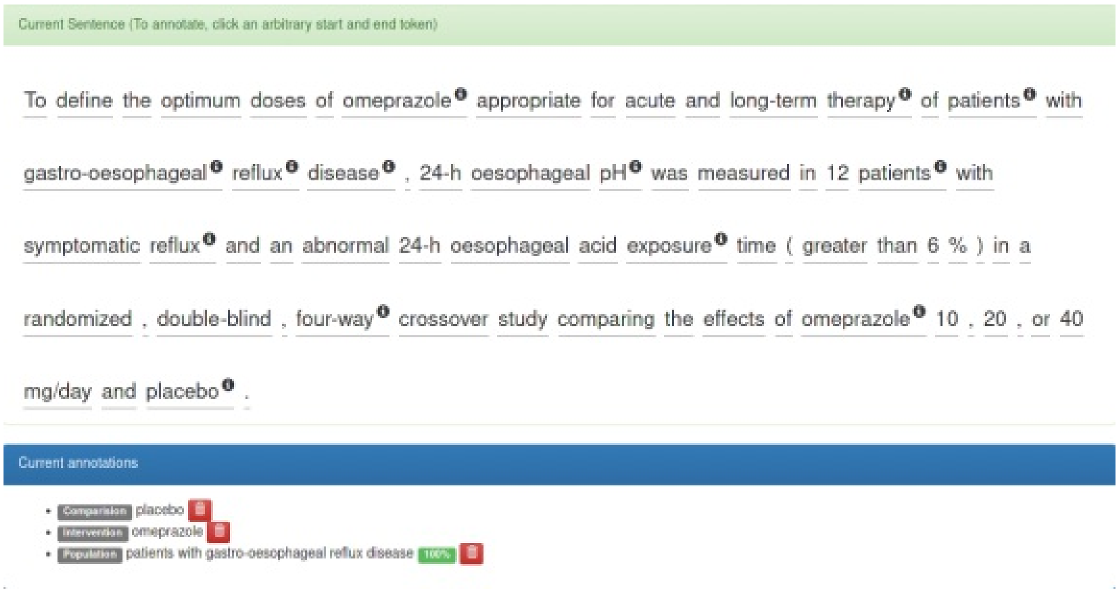


Figure 3.2: Category overview

Normally the system automatically chooses three random categories for a new user, but to make the experience of the test users comparable we chose the same category for each user.


Current Sentence (To annotate, click an arbitrary start and end token)

To define the optimum doses of omeprazole appropriate for acute and long-term therapy of patients with gastro-oesophageal reflux disease , 24-h oesophageal pH was measured in 12 patients with symptomatic reflux and an abnormal 24-h oesophageal acid exposure time ( greater than 6 % ) in a randomized , double-blind , four-way crossover study comparing the effects of omeprazole 10 , 20 , or 40 mg/day and placebo .

Current annotations

- Comparison placebo
- Intervention omeprazole
- Population patients with gastro-oesophageal reflux disease 100%

Figure 3.3: Current annotations (first version)

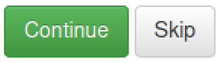


Figure 3.4: Buttons if nothing annotated (first version)

The annotation process is done in learning sessions. A session consists of up to 20 sentences of a chosen category. The sentences are randomly chosen, but sentences with existing annotations are preferred to enhance the chance to be able to give feedback for an annotation. In total 174 annotations were added by the test users. An annotation consists of a word span and a (PICO) label. For each session sentence the abstract is shown for better context. The user is able to annotated Population, Intervention and Comparison. For each annotation of a sentence the annotation is compared to the existing annotations done by other users (Section 3.2.2) and shown in a list below the sentence (Figure 3.3).

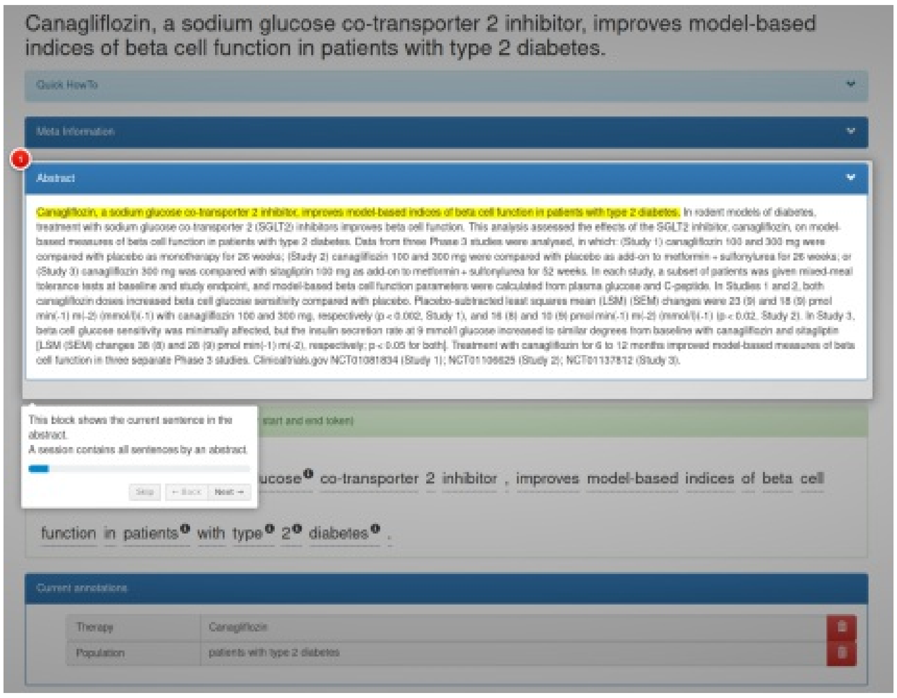
If there are no annotations in a sentence the user can continue without annotating elements (Figure 3.4). This case is also stored as a negative test case. If the user is unsure, it is possible to skip the sentence (Figure 3.4), the sentence will eventually appear in a future session. If an user annotates all sentences of a category a new category is unlocked and the finished category is marked in the overview.

After the test round we gathered the feedback (Section 4.2) of the test users.

### Second version

For the second test round we improved the elearning system according to the feedback we got after the first test round (Section 4.2). The second test round was done by four students and one linguistic expert. The same categories as in the first test round were assigned to the users.

The get-started guide was improved by adding more examples and providing an interactive step-by-step guide (Figure 3.5) how to annotate a sentence.


Canagliflozin, a sodium glucose co-transporter 2 inhibitor, improves model-based indices of beta cell function in patients with type 2 diabetes.

Canagliflozin, a sodium glucose co-transporter 2 inhibitor, improves model-based indices of beta cell function in patients with type 2 diabetes. In rodent models of diabetes, treatment with sodium glucose co-transporter 2 (SGLT2) inhibitors improves beta cell function. This analysis assessed the effects of the SGLT2 inhibitor, canagliflozin, on model-based measures of beta cell function in patients with type 2 diabetes. Data from three Phase 3 studies were analysed, in which: (Study 1) canagliflozin 100 and 300 mg were compared with placebo as monotherapy for 26 weeks; (Study 2) canagliflozin 100 and 300 mg were compared with placebo as add-on to metformin + sulfonylurea for 26 weeks; or (Study 3) canagliflozin 300 mg was compared with sitagliptin 100 mg as add-on to metformin + sulfonylurea for 52 weeks. In each study, a subset of patients was given mixed-meal tolerance tests at baseline and study endpoint, and model-based beta cell function parameters were calculated from plasma glucose and C-peptide. In Studies 1 and 2, both canagliflozin doses increased beta cell glucose sensitivity compared with placebo. Placebo-subtracted least squares mean (LSM) (SEM) changes were 23 (9) and 18 (9) pmol min(-1) m(-2) (mmol/l)(-1) with canagliflozin 100 and 300 mg, respectively (p < 0.002, Study 1), and 16 (8) and 10 (9) pmol min(-1) m(-2) (mmol/l)(-1) (p < 0.02, Study 2). In Study 3, beta cell glucose sensitivity was minimally affected, but the insulin secretion rate at 9 mmol/l glucose increased to similar degrees from baseline with canagliflozin and sitagliptin [LSM (SEM) changes 38 (8) and 28 (9) pmol min(-1) m(-2), respectively; p < 0.05 for both]. Treatment with canagliflozin for 6 to 12 months improved model-based measures of beta cell function in three separate Phase 3 studies. Clinicaltrials.gov NCT01081834 (Study 1); NCT01106625 (Study 2); NCT01137812 (Study 3).

This block shows the current sentence in the abstract.
A session contains all sentences by an abstract.

| | |
|---|---|
| Therapy | Canagliflozin |
| Population | patients with type 2 diabetes |

Quick HowTo
Meta Information
Abstract
1
Skip
← Back
Next →
start and end token)
ucose co-transporter 2 inhibitor , improves model-based indices of beta cell
function in patients with type 2 diabetes .
Current annotations

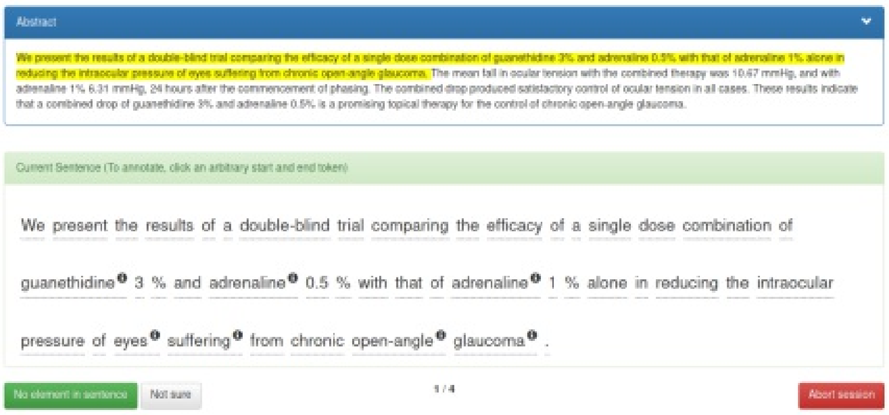
Figure 3.5: Step-by-step annotation guide

A session now consists of all sentences of a random document in sequential order as users got confused because the shown abstract (Figure 3.6) changed after each sentence to the abstract containing the new sentence.

The score of current annotations (Section 3.2.2) are hidden to prevent influencing the user to give a specific answer. Instead a session result page was introduced after all annotations are done (Figure 3.7).

A note was added to explain the correctness value as the similarity to other annotations and does not always have to be correct. The buttons for continue without annotations and skipping were renamed to make the annotation more intuitive (see Figure 3.8).

As an additional method of gamification a leaderboard was added to the application (see Figure 3.9).



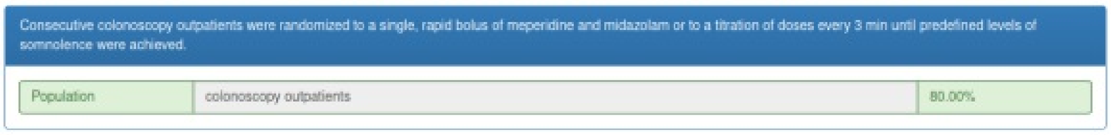
Abstract

We present the results of a double-blind trial comparing the efficacy of a single dose combination of guanethidine 3% and adrenaline 0.5% with that of adrenaline 1% alone in reducing the intraocular pressure of eyes suffering from chronic open-angle glaucoma. The mean fall in ocular tension with the combined therapy was 10.67 mmHg, and with adrenaline 1% 6.31 mmHg, 24 hours after the commencement of phasing. The combined drop produced satisfactory control of ocular tension in all cases. These results indicate that a combined drop of guanethidine 3% and adrenaline 0.5% is a promising topical therapy for the control of chronic open-angle glaucoma.

Current Sentence (To annotate, click an arbitrary start and end token)

We present the results of a double-blind trial comparing the efficacy of a single dose combination of guanethidine 3 % and adrenaline 0.5 % with that of adrenaline 1 % alone in reducing the intraocular pressure of eyes suffering from chronic open-angle glaucoma .

Figure 3.6: Annotation session (second version)



Consecutive colonoscopy outpatients were randomized to a single, rapid bolus of meperidine and midazolam or to a titration of doses every 3 min until predefined levels of somnolence were achieved.

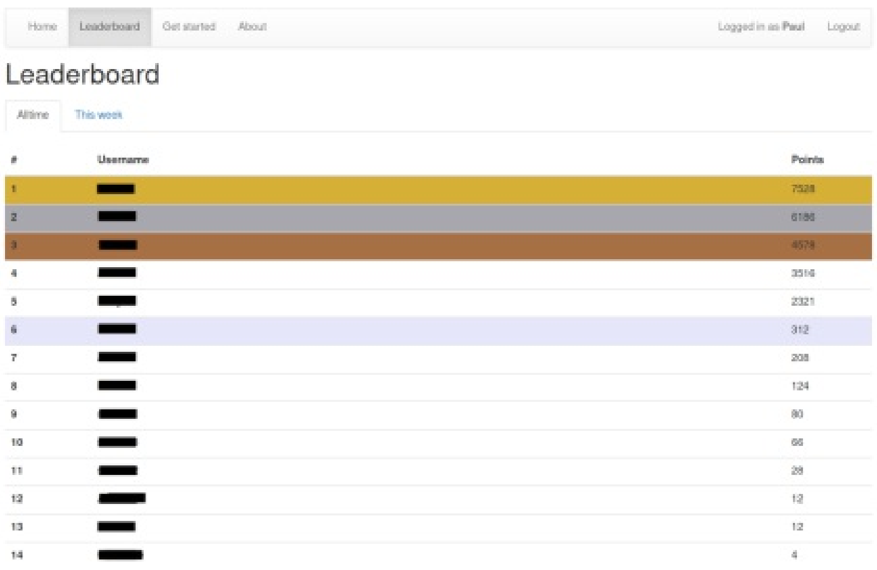
| | | |
|---|---|---|
| Population | colonoscopy outpatients | 80.00% |

Figure 3.7: Session summary page (second version)

No element in sentence
Not sure
Next sentence

Figure 3.8: Annotation buttons (second version)

Again we did a feedback session to see how well our improvements of the prototype work (Section 4.2).



Leaderboard

Alltime | This week

| # | Username | Points |
|---|---|---|
| 1 | [redacted] | 7528 |
| 2 | [redacted] | 6186 |
| 3 | [redacted] | 4578 |
| 4 | [redacted] | 3516 |
| 5 | [redacted] | 2321 |
| 6 | [redacted] | 312 |
| 7 | [redacted] | 208 |
| 8 | [redacted] | 124 |
| 9 | [redacted] | 80 |
| 10 | [redacted] | 66 |
| 11 | [redacted] | 28 |
| 12 | [redacted] | 12 |
| 13 | [redacted] | 12 |
| 14 | [redacted] | 4 |

Figure 3.9: Leaderboard (names blacked out)

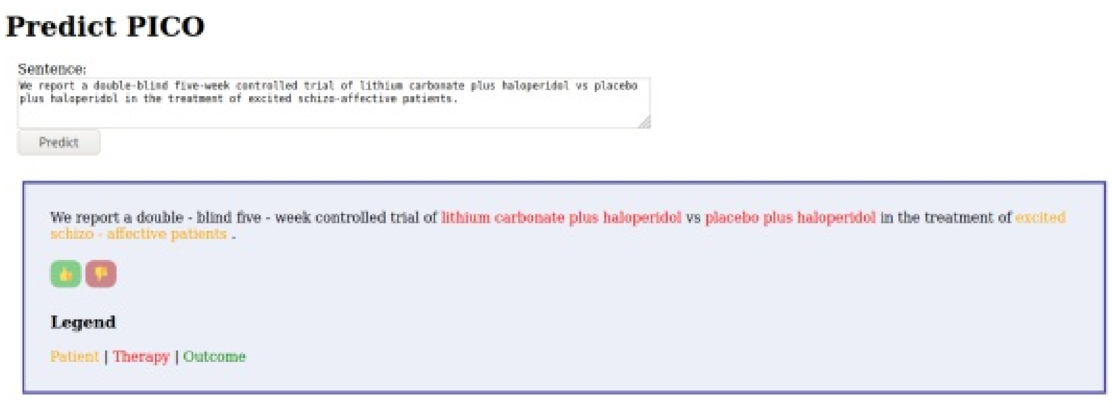
### Simplifying user feedback for predictions

After the second feedback round (Section 4.2) and the training of the SciBert model, an experimental prediction service was written allowing to run predictions by providing the text[2] or the PubMed id for the abstract[3]. The prediction service for a single sentence also allows to store a rating of the provided prediction (Figure 3.10) on the server. There exists also a rest api version[4] as an alternative for the html interface. The prediction service for an abstract (Figure 3.11) uses the provided PubMed id to retrieve the abstract using the PubMed api and run the predictor on it. As for the sentence prediction service, there exists also a rest api version[5].

This was done to allow us to manually test the output of the trained model. Additionally this api can be used in the future to provide the eLearning users with feedback on their annotation performance. The prediction rating functionality was added to allow gathering of ratings for possible future model improvements.

The eLearning system provides an admin interface we use to view and edit documents, annotations, categories, . . .
A user able to access the admin interface is called a staff user. With such a user additional features are visible in the eLearning system, for example we added a link to the service on

[2]`https://picoweb.ifs.tuwien.ac.at/pico-prediction-service/`
[3]`https://picoweb.ifs.tuwien.ac.at/pico-prediction-service/pubmed/`
[4]`https://picoweb.ifs.tuwien.ac.at/pico-prediction-service/predict/?sentence=<text>`
[5]`https://picoweb.ifs.tuwien.ac.at/pico-prediction-service/predict/pubmed/?pubmed_id=16855426`


Predict PICO

Sentence:

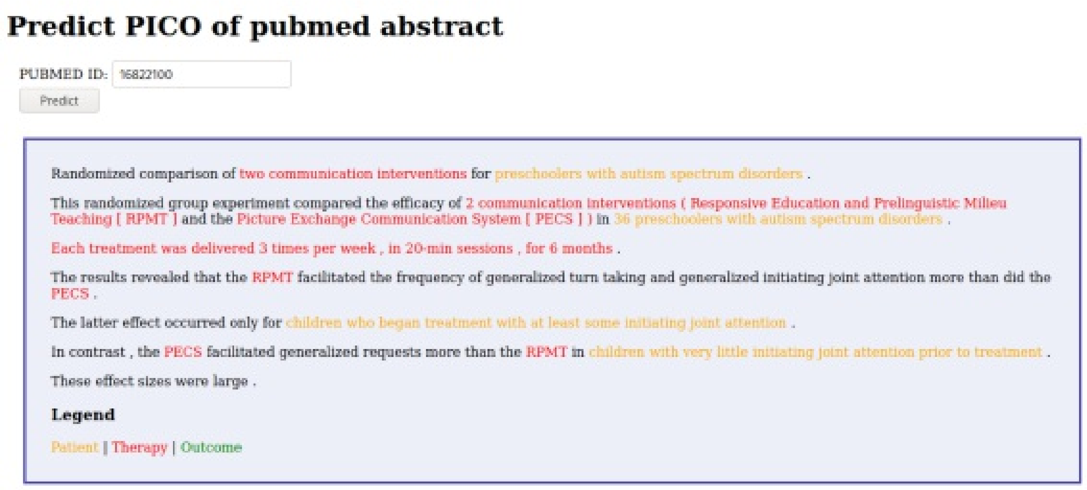
We report a double-blind five-week controlled trial of lithium carbonate plus haloperidol vs placebo plus haloperidol in the treatment of excited schizo-affective patients.

We report a double - blind five - week controlled trial of lithium carbonate plus haloperidol vs placebo plus haloperidol in the treatment of excited schizo - affective patients .

Legend

Patient | Therapy | Outcome

Figure 3.10: Prediction service for sentence


Predict PICO of pubmed abstract

PUBMED ID: 16822100

Randomized comparison of two communication interventions for preschoolers with autism spectrum disorders .

This randomized group experiment compared the efficacy of 2 communication interventions ( Responsive Education and Prelinguistic Milieu Teaching [ RPMT ] and the Picture Exchange Communication System [ PECS ] ) in 36 preschoolers with autism spectrum disorders .

Each treatment was delivered 3 times per week , in 20-min sessions , for 6 months .

The results revealed that the RPMT facilitated the frequency of generalized turn taking and generalized initiating joint attention more than did the PECS .

The latter effect occurred only for children who began treatment with at least some initiating joint attention .

In contrast , the PECS facilitated generalized requests more than the RPMT in children with very little initiating joint attention prior to treatment .

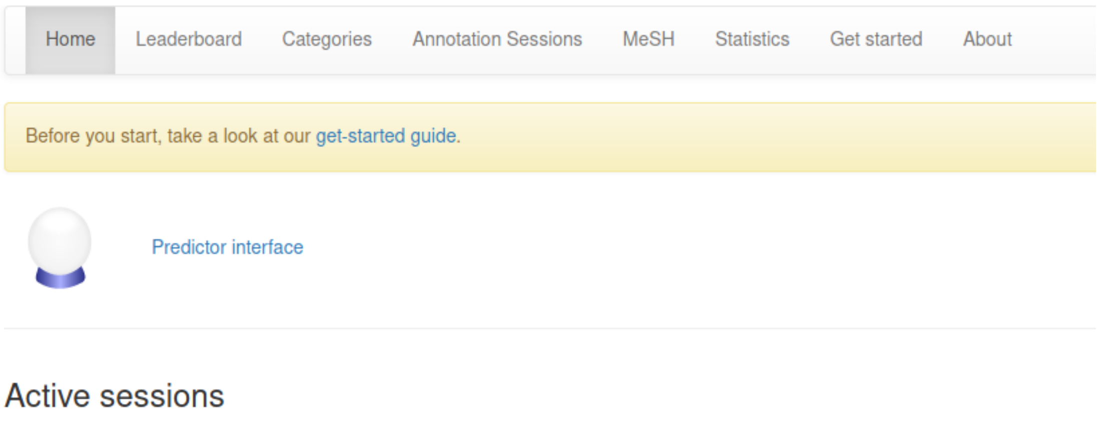
These effect sizes were large .

Legend

Patient | Therapy | Outcome

Figure 3.11: Prediction service for abstract

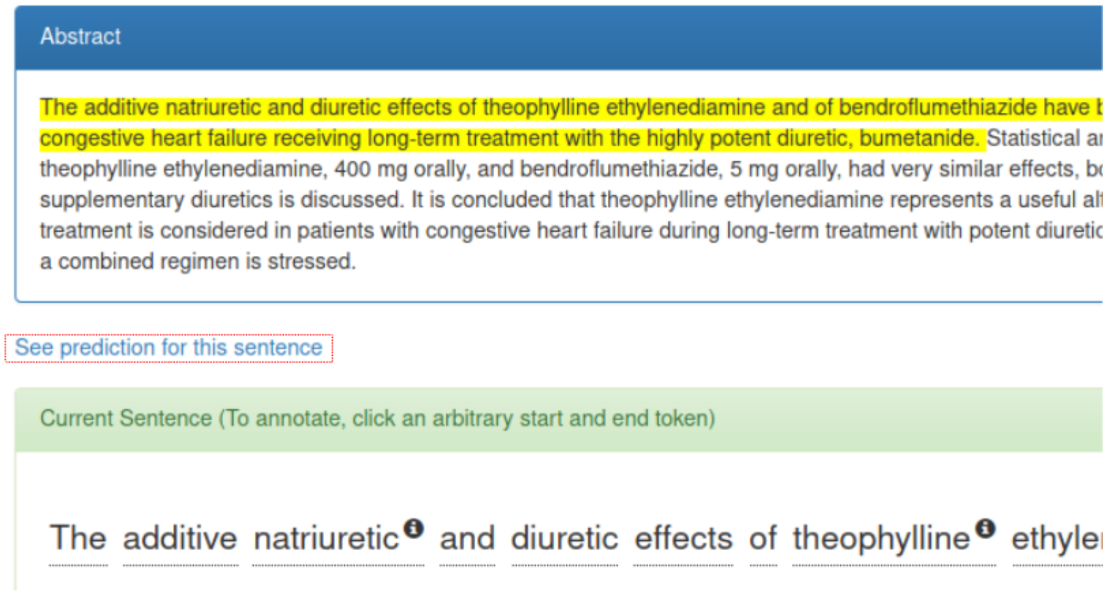
the picoweb overview page (Figure 3.12) to make it easier to test the prediction service. We also added the possibility to directly open the prediction of the current sentence inside an annotation session (Figure 3.13).



Before you start, take a look at our get-started guide.

Active sessions

Figure 3.12: Prediction service link shown on picoweb homepage for staff user



The additive natriuretic and diuretic effects of theophylline ethylenediamine and of bendroflumethiazide have b
congestive heart failure receiving long-term treatment with the highly potent diuretic, bumetanide. Statistical ar
theophylline ethylenediamine, 400 mg orally, and bendroflumethiazide, 5 mg orally, had very similar effects, bo
supplementary diuretics is discussed. It is concluded that theophylline ethylenediamine represents a useful alt
treatment is considered in patients with congestive heart failure during long-term treatment with potent diuretic
a combined regimen is stressed.

Current Sentence (To annotate, click an arbitrary start and end token)

The additive natriuretic and diuretic effects of theophylline ethyle

Figure 3.13: Sentence prediction service link (highlighted by red border) shown on annotation session view for staff user

## 3.3 Evaluation of the comparison of different data sets

To know how the SciBERT model performs on variations of the picoweb data set (Section 2.1.1) compared to the ebmnlp data set (Section 2.1.2), we trained it on these data set variations, to predict PICO labels in PubMed abstracts.

These variations are as follows:

- Only sentences containing labeled PICO elements

- Additionally including *sentences marked as not containing PICO elements*
- Adding PoS tags to the previous variation of the data set

The same preprocessing steps as in Section 3.1 were performed on the picoweb and ebmnlp data sets. CoreNLP [Man+14] was used to compute Part of Speech (PoS) tags for tokens. 75% of the annotations are used as training set.

SciBERT [BLC19] was used together with a Bidirectional Recurrent Neural Network (BRNN) to train on the picoweb and ebmnlp data sets.

We used the reference setup for SciBERT [BLC19] (commit: 7598219) published on github[6]. We mostly used the preset hyper-parameters of SciBERT without finetuning the learning rate. The used parameters are described in Table 3.2 All trainings were done on a Nvidia GTX 1060 6GB using PyTorch [Pas+19]. We exported the picoweb data set to the CoNLL-2003 format [SD03] to be loaded into the SciBERT model, as the ebmnlp data set uses this format too.

To measure the performance of the trained models we used token-level f1-scores (Section 2.5). Rather than using a fixed number of training epochs, we used an early stopping technique to train as long as the average validation f1-score increases. This was combined with a technique often called *patience* to continue training for ten epochs after the average validation f1-score decreases.

| Optimizer | AdamW/BertAdam |
|---|---|
| Encoder | bidirectional LSTM |
| Layers | 2 |
| Layer size | 200 |
| Dropout | 0.5 |
| Learning rate | 0.001 |
| Batch size | 64 |
| Dropout | 0.5 |
| Validation metric | average f1-score |

Table 3.2: SciBERT model parameters

Besides sentences containing PICO elements we also gave the user the possibility to explicitly mark sentences that do not contain any PICO labels. This decision based on two ideas. Firstly, it provides us with more data to learn from. Secondly, this provides us with (more) examples of words that are not PICO elements. There are new approaches of binary classification without negative data like positive-confidence (Pconf) [INS18] where they use confidence information besides the positive data. Using the right amount of negative examples can have a big impact on the performance of the model [KSB14]. We

[6]https://github.com/allenai/scibert (visited on 2020-03-22)

trained the SciBERT model on our data set with and without including sentences not containing PICO labels.

Next we added the PoS tags we generated earlier using CoreNLP [Man+14] to the CoNLL-2003 format [SD03] export. This was done to answer the question if including predicted PoS tags can improve the performance. To answer the question how the picoweb data set performs compared to a similar data set, we computed the model on the ebmnlp data set included in SciBERT [BLC19][7].

More training data in general lead to a better model performance. As the ebmnlp data set is bigger than the picoweb data set, we use a similar sized random subsample of ebmnlp to compensate for the impact a bigger data set has on the model performance. Additionally we removed all annotations for *Outcome* from the ebmnlp data set, to enhance comparability between the picoweb and ebmnlp data set, as the picoweb data set does not contain annotations for *Outcome*. Last we train on a combination of both data sets to explore if this leads to a better model performance. The combining was done by concatenating both data sets in the CoNLL-2003 format [SD03]. Care has been taken to ensure that annotations of the same sentence are not used in both training and validation.

## 3.4 Summary

In this chapter we shown how we applied the algorithms described in the background to to get rudimentary insights into the data sets. Furthermore, we presented which design principles were implemented for the eLearning system. Additionally we described the created categories based on MeSH tags. At last we presented the parameters for our Machine Learning model.

[7]`https://github.com/allenai/scibert/tree/master/data/pico/ebmnlp` (visited on 2020-03-22)

# CHAPTER 4

# Result

## 4.1 Data set analysis findings

In this section we present the result of our analysis of our data set. Further, we explore the differences between our data set and the ebmnlp data set [Nye+18].

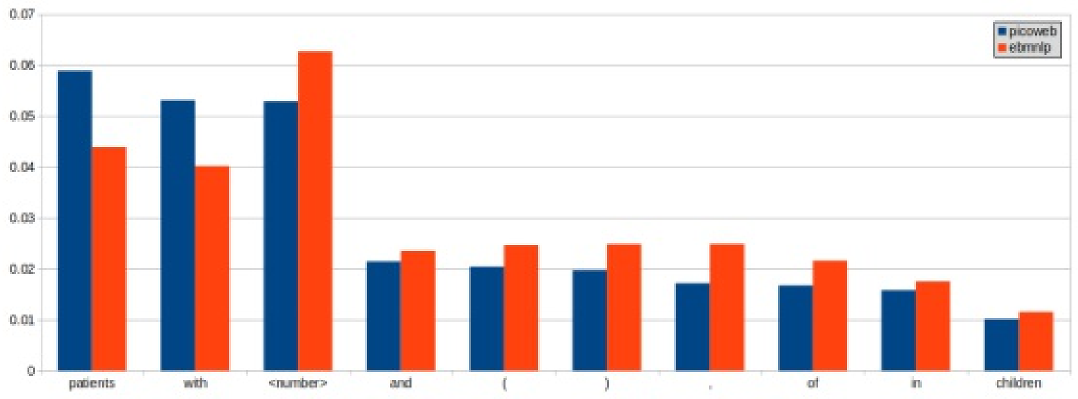
The picoweb data set (Section 2.1.1) consists of annotations of approximately 1400 documents containing approximately 9000 sentences. Overall nearly 26000 annotations were gathered, over 7600 labeled as population and over 18000 as therapy. The picoweb data set consists of over 120000 annotated tokens, where approximately 50000 labeled as population and approximately 70000 as therapy.

The ebmnlp data set (Section 2.1.2) in comparison consists of 5000 annotated documents. We could not find the annotator guidelines that were used to create the ebmnlp data set. Therefore we concentrated on analyzing the annotations itself.

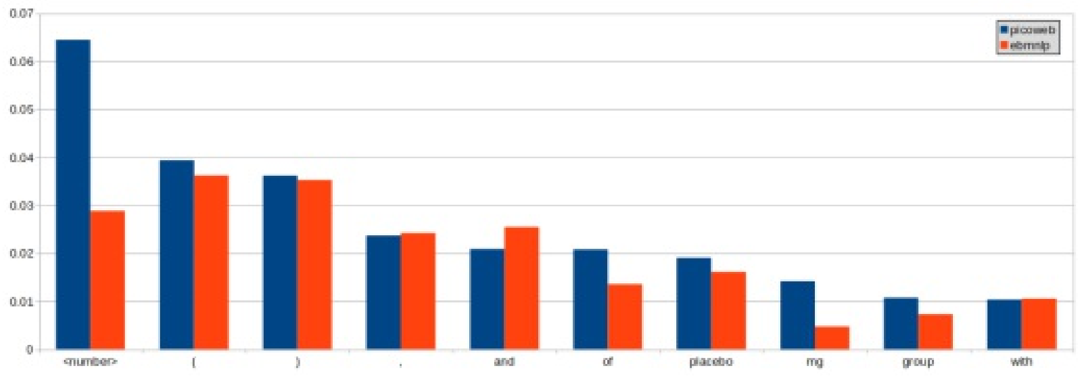
First, we want to know how similar the picoweb and the ebmnlp dataset are. Therefore we calculated the cosine similarity as stated in Section 3.1.

For the picoweb data set the cosine similarity (Section 3.1) between therapy and population is 0.648. In contrast for the ebmnlp data set the cosine similarity between therapy and population is 0.779. This shows that the labeled tokens for population and therapy differ less inside the ebmnlp compared to the picoweb data set.

Next we wanted to see what the most common words of both data sets are and how the differ. Figure 4.1 shows the frequency of the ten most annotated words in the picoweb data set for each label. For the same words we also show the frequency of annotations in the ebmnlp data set. One can see that 6% of the *Population* annotations in the picoweb data set is the word *patients*, compared to 4% in the ebmnlp data set. For *Therapy* the most commonly annotated words were numbers in general with 6.5% in picoweb, compared to 3% in ebmnlp. Together with the difference in the annotation of *mg* ( 1.5% vs. 0.5%) this indicates that annotators of the picoweb data set included the dosage of a drug more often than in the ebmnlp data set.



(a) population



(b) therapy

Figure 4.1: relative word frequencies by annotated label

Both data sets have 17 annotated documents in common. We picked one containing intervention with dosage to see the difference. While the ebmnlp example (see Figure 4.2) contains multiple smaller precise annotations, the picoweb example (see Figure 4.3) consists of one large annotation catching the whole therapy procedure.

After a screening nocturnal polysomnograms ( NPSG ) and MSLT the following day , participants with primary insomnia were randomized to take zolpidem 10 mg ( n = 50 ) or placebo ( n = 45 ) nightly for 12 months .

Figure 4.2: ebmnlp example population(orange)/therapy(red)

After a screening nocturnal polysomnograms ( NPSG ) and MSLT the following day , participants with primary insomnia were randomized to take zolpidem 10 mg ( n = 50 ) or placebo ( n = 45 ) nightly for 12 months .

Figure 4.3: picoweb example therapy(red)

We calculated the entropy (Section 3.1) of the likelihood being labeled *population* or *therapy* of the most common labeled words in picoweb and ebmnlp. This was done to estimate how the SciBERT model performs when combining both data sets. High entropy means the word is labeled as both *Population* and *Therapy* equally often. On the other hand a low entropy means the word clearly belongs to one of the two categories.

| **token** | **entropy** | **population count** | **therapy count** | **unlabeled count** |
|---|---|---|---|---|
| failure | 0.5692148 | 163 | 0 | 76 |
| adults | 0.3607534 | 147 | 0 | 23 |
| hypertensive | 0.2610894 | 77 | 0 | 7 |
| young | 0.2167368 | 73 | 0 | 5 |
| japanese | **0.1544296** | 71 | 0 | 3 |
| section | 0.4176604 | 53 | 0 | 11 |
| colorectal | 0.4554859 | 48 | 0 | 12 |
| forty | 0.5268998 | 47 | 0 | 17 |
| arthritis | 0.5268998 | 47 | 0 | 17 |
| adolescents | 0.3510629 | 47 | 0 | 7 |

Table 4.1: picoweb entropy for words most commonly labeled *Population* in picoweb data set

| **token** | **entropy** | **population count** | **therapy count** | **unlabeled count** |
|---|---|---|---|---|
| failure | 0.6662032 | 191 | 7 | 302 |
| adults | 0.4061009 | 365 | 2 | 63 |
| hypertensive | 0.4496421 | 143 | 1 | 12 |
| young | 0.4580312 | 214 | 2 | 45 |
| japanese | 0.5666924 | 40 | 3 | 7 |
| section | 0.8365317 | 19 | 4 | 25 |
| colorectal | 0.7640933 | 83 | 8 | 94 |
| forty | 0.2537468 | 69 | 0 | 6 |
| arthritis | 0.5224450 | 68 | 0 | 24 |
| adolescents | 0.3997988 | 164 | 3 | 22 |

Table 4.2: ebmnlp entropy for words most commonly labeled *Population* in picoweb data set

We first calculated the entropy for the 10 words most often labeled *Population* in picoweb (Table 4.1). As we can see even if the word *failure* is never annotated as *Therapy* the entropy is high with 0.5692148, as there are 76 unlabeled occurrences of the word *failure*. With 0.1544296 the word *japanese* has the lowest entropy, this means appearances of this word are most likely labeled *Population*. For comparison we calculated the entropy for the ebmnlp data set for the same words (Table 4.2). For eight of the words the entropy is higher in the ebmnlp data set. The entropy of the word *arthritis* is with 0.5224450 very similar to the picoweb entropy of 0.5268998. For the word *japanese* the entropy is with 0.5666924 much higher for ebmnlp than for picoweb with 0.1544296. The word *forty* on the other hand has a much lower entropy of 0.2537468 for ebmnlp than picoweb entropy with 0.5268998.

| **token** | **entropy** | **population count** | **therapy count** | **unlabeled count** |
|---|---|---|---|---|
| disorder | 0.5587121 | 398 | 0 | 173 |
| hundred | 0.2495566 | 236 | 0 | 20 |
| older | 0.5563372 | 177 | 0 | 76 |
| metastatic | 0.5310091 | 127 | 0 | 47 |
| males | 0.4386603 | 113 | 0 | 26 |
| infarction | 0.6156223 | 105 | 0 | 152 |
| leukemia | 0.5000691 | 83 | 0 | 26 |
| previously | 0.6058537 | 82 | 0 | 132 |
| females | 0.5188793 | 78 | 0 | 27 |
| old | 0.3124197 | 74 | 0 | 9 |

Table 4.3: ebmnlp entropy for words most commonly labeled *Population* in ebmnlp data set

| **token** | **entropy** | **population count** | **therapy count** | **unlabeled count** |
|---|---|---|---|---|
| disorder | 0.5487493 | 96 | 3 | 24 |
| hundred | 0.3313479 | 119 | 2 | 11 |
| older | 0.4979830 | 105 | 1 | 27 |
| metastatic | 0.2468472 | 36 | 0 | 3 |
| males | 0.4729420 | 33 | 0 | 9 |
| infarction | 0.5857992 | 111 | 3 | 36 |
| leukemia | 0.6881253 | 15 | 1 | 6 |
| previously | 0.7800730 | 24 | 3 | 30 |
| females | 0.4781681 | 25 | 0 | 7 |
| old | 0.3084670 | 42 | 0 | 5 |

Table 4.4: picoweb entropy for words most commonly labeled *Population* in ebmnlp data set

Afterwards we calculated the entropy for the 10 words most often labeled *Population* in ebmnlp (Table 4.3). There are only two words with an entropy less than 0.4, *hundred* with an entropy of 0.2495566 and *old* with 0.3124197. For comparison we calculated the entropy for the picoweb data set for the same words (Table 4.4). Similar to the entropies for picoweb, in the ebmnlp data set the entropies for the words *hundred* (0.3313479) and *old* (0.3084670) are relatively low. In contrast to the entropy of 0.5310091 for the word *metastatic* in the picoweb data set, the entropy is much lower in the ebmnlp data set with 0.2468472.

| **token** | **entropy** | **population count** | **therapy count** | **unlabeled count** |
|---|---|---|---|---|
| twice | 0.4733274 | 0 | 172 | 47 |
| administration | 0.6218378 | 0 | 140 | 186 |
| solution | 0.3777006 | 0 | 135 | 23 |
| saline | 0.3190870 | 0 | 119 | 15 |
| lidocaine | 0.2509362 | 0 | 117 | 10 |
| times | 0.6137725 | 0 | 111 | 75 |
| administered | 0.6245256 | 0 | 93 | 118 |
| low-dose | 0.3429510 | 0 | 84 | 12 |
| losartan | 0.3454873 | 0 | 76 | 11 |
| sertraline | 0.3454873 | 0 | 76 | 11 |

Table 4.5: picoweb entropy for words most commonly labeled *Therapy* in picoweb data set

| **token** | **entropy** | **population count** | **therapy count** | **unlabeled count** |
|---|---|---|---|---|
| twice | 0.6271984 | 11 | 67 | 214 |
| administration | 0.4283185 | 21 | 78 | 623 |
| solution | 0.6485577 | 2 | 128 | 79 |
| saline | 0.4861115 | 1 | 192 | 50 |
| lidocaine | 0.3282946 | 0 | 136 | 18 |
| times | 0.3790360 | 11 | 35 | 359 |
| administered | 0.4506919 | 19 | 65 | 489 |
| low-dose | 0.7060968 | 4 | 57 | 88 |
| losartan | 0.5353743 | 0 | 29 | 11 |
| sertraline | 0.4205641 | 0 | 19 | 4 |

Table 4.6: ebmnlp entropy for words most commonly labeled *Therapy* in picoweb data set

The same calculations as for *Population* were done for *Therapy*. We calculated the entropy for the 10 words most often labeled *Therapy* in picoweb (Table 4.5). For comparison we calculated the entropy for the ebmnlp data set for the same words (Table 4.6). The only word with low entropy in both data sets is *lidocaine* with 0.2509362 in the picoweb data set and 0.3282946 in the ebmnlp data set.

| token | entropy | population count | therapy count | unlabeled count |
|---|---|---|---|---|
| risperidone | 0.4193760 | 0 | 215 | 45 |
| bupivacaine | 0.3581505 | 0 | 136 | 21 |
| lidocaine | 0.3282946 | 0 | 136 | 10 |
| cyclophosphamide | 0.1452165 | 0 | 106 | 4 |
| oil | 0.5780314 | 0 | 105 | 52 |
| doxorubicin | 0.2923156 | 0 | 103 | 11 |
| morphine | 0.6147689 | 0 | 98 | 67 |
| oxytocin | 0.5315167 | 0 | 97 | 36 |
| cbt | 0.4338554 | 0 | 89 | 20 |
| propranolol | 0.4007739 | 0 | 89 | 17 |

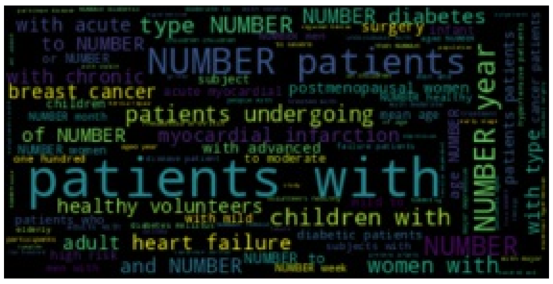
Table 4.7: ebmnlp entropy for words most commonly labeled *Therapy* in ebmnlp data set

| token | entropy | population count | therapy count | unlabeled count |
|---|---|---|---|---|
| risperidone | 0 | 0 | 32 | 0 |
| bupivacaine | 0.4938054 | 2 | 98 | 21 |
| lidocaine | 0.2509362 | 0 | 117 | 10 |
| cyclophosphamide | 0.6506238 | 1 | 29 | 13 |
| oil | 0.6098685 | 0 | 17 | 11 |
| doxorubicin | 0.5240761 | 1 | 32 | 7 |
| morphine | 0.8527925 | 7 | 35 | 21 |
| oxytocin | 0.6216097 | 0 | 3 | 4 |
| cbt | 0.4695343 | 3 | 22 | 1 |
| propranolol | 0.5261106 | 1 | 35 | 8 |

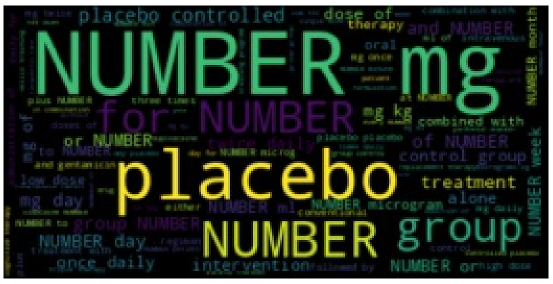
Table 4.8: picoweb entropy for words most commonly labeled *Therapy* in ebmnlp data set

As before we calculated the entropy for the 10 words most often labeled *Therapy* in ebmnlp (Table 4.7). There are several drug names with low entropy, especially the words *cyclophosphamide* and *lidocaine* have low entropies of 0.1452165 respectively 0.3282946. For comparison we calculated the entropy for the picoweb data set for the same words (Table 4.8). For the ebmnlp data set only two words have low entropies. The word *lidocaine* with 0.2509362 and the word *risperidone* with 0, as it is always annotated as therapy and the shannon entropy defines $log(0)$ as 0 (Section 2.1.3).

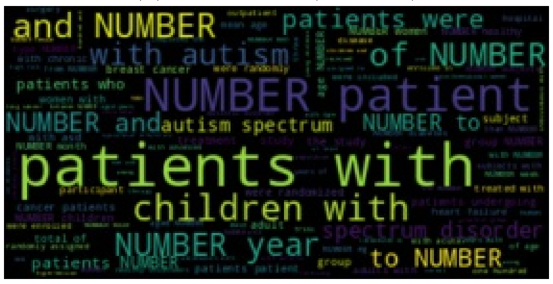
In contrast to the top 10 words of the picoweb data set (Table 4.5) the top 10 words of the ebmnlp data set (Table 4.7) only contain pharmaceutical ingredients. This shows us that ebmnlp annotations tend to be shorter and only contain the most essential elements (Figure 4.2).



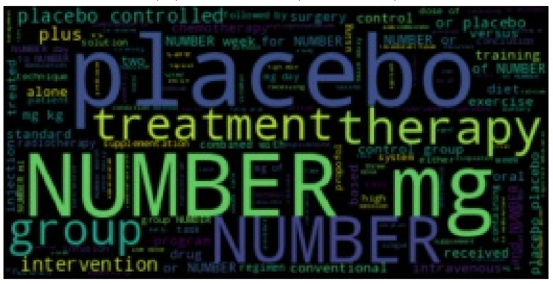
(a) population (picoweb)



(b) therapy (picoweb)



(c) population (ebmnlp)



(d) therapy (ebmnlp))

Figure 4.4: word cloud by annotated label

We used the *word_cloud* library for python[1] to generate word clouds for picoweb and ebmnlp, both for population and therapy separately. Comparing Figure 4.4a and Figure 4.4c we see a lot of similarities in *Population*, like *patients with* and *<number> patient*. There are only differences in some detailed diseases, like for picoweb there are more annotations of breast cancer and diabetes patients, while for ebmnlp autism is more frequent.
Doing the same for therapy Figure 4.4b and Figure 4.4d we see that *treatment* and *therapy* is more frequent in the ebmnlp data set compared to the picoweb data set.

Besides the high rate of the word *patients*, Figure 4.4a shows that the label *population* is highly likely linked to a disease.

Figure 4.4b confirms the insight that, as we only use clinical trials in the data set, control groups (*placebo*) are omnipresent in *therapy* annotations.

Looking at the lengths of the annotations by label (Figure 4.5), one can see that annotations of population tend to include more tokens than therapy. If we look at some samples of annotations, therapy often just consists of a drug name (*e.g. dexamethasone* or *placebo*). Sometimes followed by the dosage (*e.g. full-dose ( 10 mg ) ramipril*). While population often contains a group of people combined with a disease (*e.g. infants and children with uncomplicated Salmonella gastroenteritis*). Some annotators also included the count of the patient group (*e.g. 108 patients with angina*).

[1]python word cloud library: `https://github.com/amueller/word_cloud` (visited on 2020-04-18)

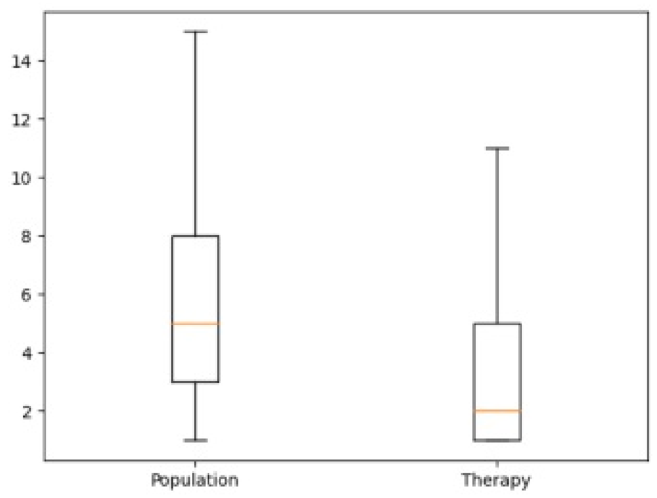


Figure 4.5: Length of annotated sequence in picoweb

## 4.2 Gamification/eLearning platform

The feedback sessions after the test rounds (Section 3.2.3) gave us valuable insights into how to make the system as easy to use as possible, without sacrificing too much annotation quality.

After the first test round was completed (Section 3.2.3) there was a feedback round. Following points were discussed, evaluated and if possible improved in the second prototype. Seeing the abstract of the sentence leads to the wish to select another sentence in the abstract with better PICO elements in it. However cherry picking the best sentences from an abstract leads to the problem of having only positive examples. This probably result in a neural network not "knowing" sentences without PICO elements can exist. To improve the usability here, while preventing cherry-picking, we came up with the idea to change an annotation session to consist of a complete abstract (Section 3.2.3). By leading the user through the abstract sentence by sentence, the context of the current sentence should be easier to understand. The feedback also showed that the distinction between Intervention and Comparison is often difficult to see. This is especially true for studies comparing three or more therapies. Also it probably does not matter most of the time if a paper compares therapy X with therapy Y or the other way around.
We noticed that the *correctness* shown for annotations can lead to the assumption that 100% are "the truth". To counteract this we added an info text next to the *correctness* explaining what it represents. Furthermore the get-started guide needs more examples of how to annotate specific elements as it is not likely eLearning users take the time to read the full annotation guideline. A common question was if you should annotate the amount of a population or not (*e.g. 50 patients with heart disease* vs. *patients with heart*

*disease*).

Similar to the first test round, we collected the feedback of the test users after the second test round (Section 3.2.3). The step-by-step guide received good feedback, the users are now less confused how to use the platform. The same is valid for the new session layout, where all sentences of a whole abstract are used for a session. One user noted that the *No element in sentence* button does not need confirmation, so it is possible to trigger it by accident, without a possibility to undo it. This is not fixed yet.

To answer the question on how to evaluate the performance of users, we used Fleiss' Kappa to calculate the IAA to be used to give the users feedback on their performance. As our test rounds did not go over a long period of time, showing the user the IAA (Section 3.2.2) of the annotations done in a session was sufficient feedback for our test users. With a long-term user in mind who keeps improving, some kind of overall user score will be necessary to be implemented in the future, to give the user clear feedback of his/her progress.

For our tests we created the categories by hand using MeSH tags (Section 3.2.1). This was possible as we only created four simple categories by naively assigning MeSH tags containing specific keywords (*e.g. heart*, *cardiac*, ...). For more complex categories the MeSH tags should be provided as a tree structure to see what tags are related.

## 4.3 Comparing the SciBERT model performance trained on different data sets with each other

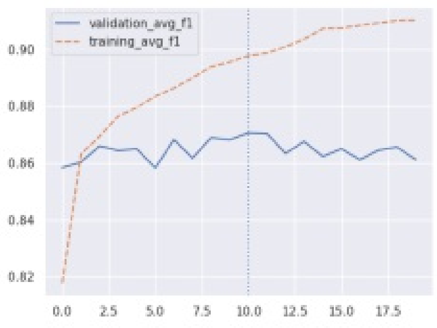
In this section we wanted to find out how the SciBERT model (Section 3.3) performs using variations of the picoweb data set as input. The SciBERT model was trained to predict the labels *Population*, *Therapy*, and *Outside* (representing non-PICO elements). We also trained the SciBERT model on the ebmnlp data set (Section 2.1.2) to compare both performances.

We started by exploring the impact of including sentences without PICO elements on the performance of the SciBERT model. Therefore we trained on an export of our annotations only including sentences with PICO labels and an export also including sentences without labels to compare their performance (Section 4.3.1). Next we explored the impact of PoS tags on the model performance (Section 4.3.2).

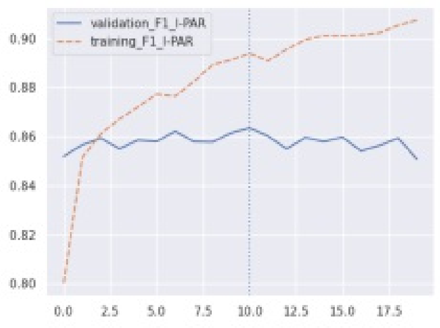
### 4.3.1 Evaluating the influence of storing sentences without PICO labels on the SciBERT model performance

To evaluate the impact of using sentences without any PICO labels on the model performance, we trained the SciBERT model [BLC19] one time using the picoweb data set excluding *sentences marked as not containing PICO elements* and another time including *sentences marked as not containing PICO elements*. Overall there were 25820 annotated sentences with annotated PICO elements, but only 395 that are marked by the user as not containing any PICO elements. This is because most of the annotations derived from a former project (see: Section 2.1.1), where it was not possible to mark sentences that do not contain PICO elements.

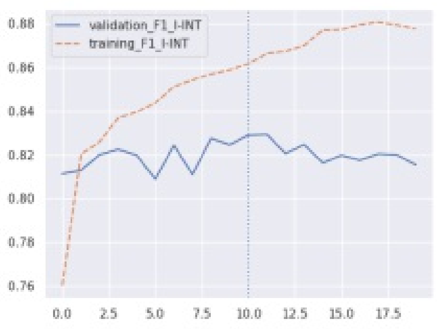
Each training ran until the avg f1-score did not improve for 10 epochs. Best scores were achieved after epoch 6 for the set including *sentences marked as not containing PICO elements*. In contrast it took until epoch 10 for the set not including *sentences marked as not containing PICO elements* to reach the best f1 score.



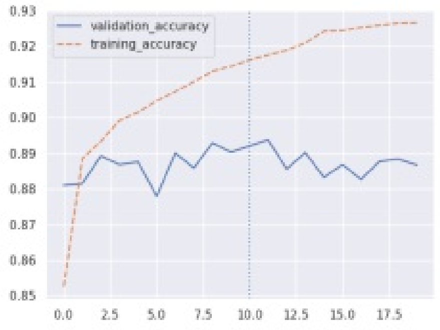
(a) avg f1-scores



(b) f1-scores population



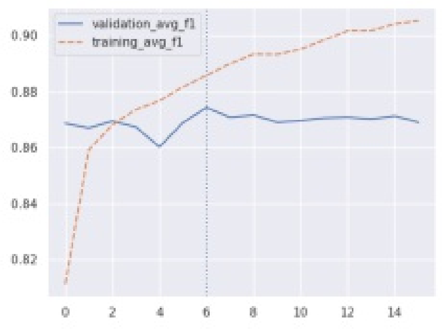
(c) f1-scores therapy



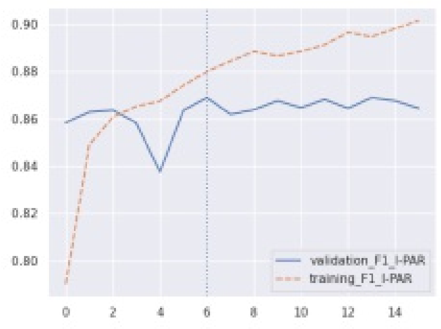
(d) accuracy

Figure 4.6: f1-scores and accuracy for picoweb data set excluding *sentences marked as not containing PICO elements*

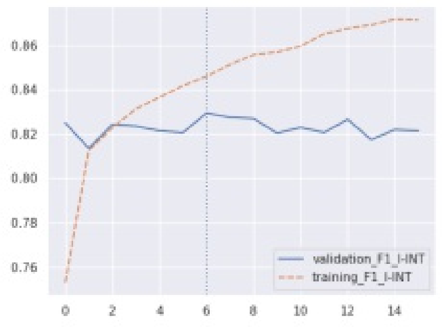
Figure 4.6 shows the training process of the model excluding *sentences marked as not containing PICO elements*. As we can see in Subfigure 4.6b the validation f1-score for *Population* is even through the epochs, this seems most likely based on the consistency of *Population* annotations (for example *patients*). In comparison the f1-scores for *Therapy* are more uneven (Subfigure 4.6c), very likely because of the diverse possibilities in *Therapy* (*e.g. medication*, *surgery*, ...). After epoch 10 the validation scores beginning to fall while the training scores still increase, this indicates overfitting. Therefore, we choose the state after epoch 10.



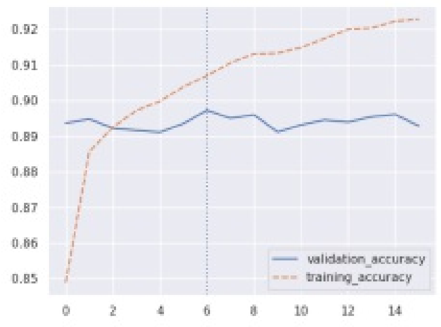
(a) avg f1-scores



(b) f1-scores population



(c) f1-scores therapy



(d) accuracy

Figure 4.7: f1-scores and accuracy for picoweb data set including *sentences marked as not containing PICO elements*

Figure 4.7 shows the training process of the model including *sentences marked as not containing PICO elements*. After epoch 6 the validation scores begin to fall while the training scores still increase, this indicates overfitting. Therefore, we choose the state after epoch 6. In contrast to the training process of the data set including *sentences marked as not containing PICO elements*—except for the most likely random outlier at epoch 4—the validation scores stay more stable after the best epoch.

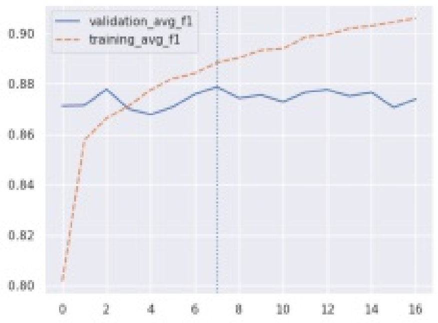
| | **Population** | **Therapy** | **Outside** | **Average** |
|---|---|---|---|---|
| **without negatives** | 0.86 | 0.83 | 0.92 | 0.87 |
| **with negatives** | **0.87** | 0.83 | 0.92 | 0.87 |

Table 4.9: f1-scores after 6 epochs for picoweb not including *sentences marked as not containing PICO elements* and 10 epochs including *sentences marked as not containing PICO elements*

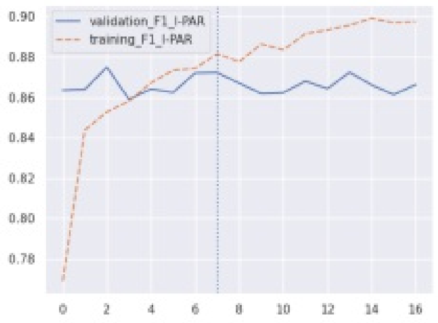
As shown in Table 4.9 including *sentences marked as not containing PICO elements* result in a minimal higher (less than 0.01) f1-score for *Population*.

### 4.3.2 Evaluating the impact of PoS tags

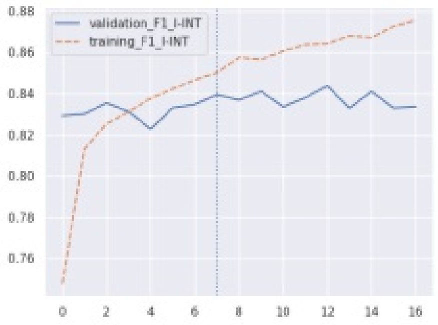
In this section we evaluate the possible benefit of adding PoS tags into the picoweb data set. The PoS tags were created using CoreNLP [Man+14] and added to the corresponding fields in the CoNLL-2003 [SD03] export file. We expected an increase of the performance based on the idea that the PoS tags provide additional indications for PICO labels. Surprisingly, adding PoS tags only led to less than 0.01 higher average f1-score (Table 4.10). One explanation for this effect is, that the SciBERT model already learned similar categories like the PoS in the pre-training done on corpus consisting of computer science and biomedical papers.



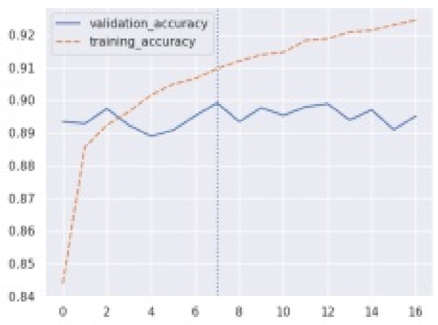
(a) avg f1-scores



(b) f1-scores population



(c) f1-scores therapy



(d) accuracy

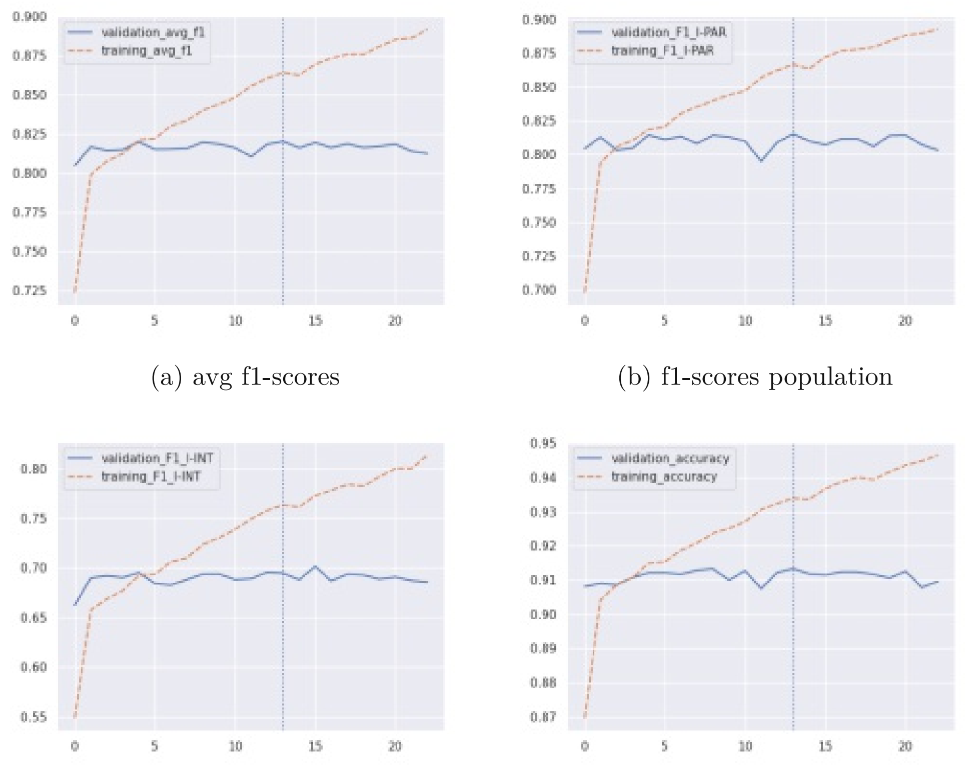
Figure 4.8: f1-scores and accuracy for picoweb data set with PoS tags

The training progress over the epochs seen in Figure 4.8 behaved similar as the data set without PoS seen in Figure 4.7, In contrast to the training on the data set without PoS tags—which showed overfitting after epoch 6—the data set with PoS showed overfitting after epoch 7.

| | Population | Therapy | Outside | Average |
|---|---|---|---|---|
| **with PoS tags** | 0.87 | **0.84** | 0.92 | **0.88** |
| **without PoS tags** | 0.87 | 0.83 | 0.92 | 0.87 |

Table 4.10: f1-scores for picoweb with and without PoS tags

### 4.3.3 SciBERT model performance trained on ebmnlp compared to picoweb



(a) avg f1-scores

(b) f1-scores population

(c) f1-scores therapy

(d) accuracy

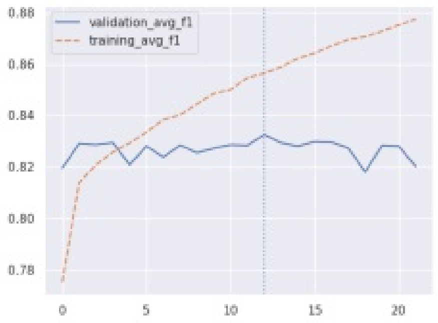
Figure 4.9: f1-scores and accuracy for ebmnlp data set downsized to similar size as picoweb and removed outcome

Figure 4.9 shows the training process of the model training on the ebmnlp data set (Section 4.3.3). After epoch 13 the validation scores begin to fall while the training scores still increasing, this indicates overfitting. Therefore, we choose the state after epoch 13. As shown in Table 4.11 the model trained on picoweb performed slightly better by about 0.05 for *Population* and substantially for *Therapy* by 0.11.

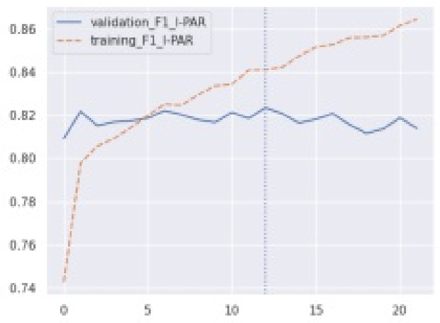
| | Population | Therapy | Outside | Average |
|---|---|---|---|---|
| **picoweb** | **0.86** | **0.83** | 0.92 | **0.87** |
| **ebmnlp** | 0.81 | 0.72 | 0.90 | 0.79 |

Table 4.11: Comparison f1-scores for ebmnlp subsample data set without *Outcome* annotations and picoweb data set excluding *sentences marked as not containing PICO elements* (Section 4.3.1) data set

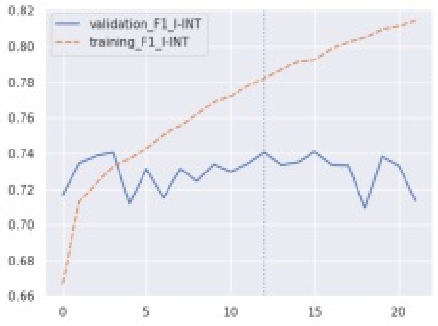
### 4.3.4 Combine picoweb and EBM-NLP



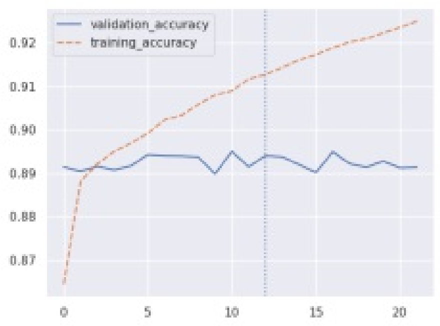
(a) avg f1-scores



(b) f1-scores population



(c) f1-scores therapy



(d) accuracy

Figure 4.10: f1-scores and accuracy for combined data sets (picoweb & ebmnlp)

Figure 4.10 shows the training process of the model for the combined data set (Section 3.3). The f1-score (Table 4.12) is slightly higher with an average of 0.83 than ebmnlp with 0.79, but lower than the f1-score of picoweb with 0.92. After epoch 12 the validation scores begin to fall while the training scores still increase, this indicates overfitting. Therefore, we choose the state after epoch 12.

| Population | Therapy | Outside | Average |
|---|---|---|---|
| 0.82 | 0.74 | 0.93 | 0.83 |

Table 4.12: f1-scores after epoch 12 for ebmnlp data set downsized to similar size as picoweb and removed outcome

## 4.4 Summary

In this chapter we presented the result of our analysis of our data set. We calculated the cosine similarity between *Population* and *Therapy* for the picoweb (0.648) and ebmnlp (0.779) data set. This showed us that in ebmnlp the words labeled *Population* and *Therapy* are more similar than in picoweb, which probably makes it harder to predict these labels in ebmnlp. Another important difference is that in ebmnlp mostly does not recognize units of dosages as part of *Therapy* as shown in Figure 4.1. We calculated the entropy of the likelihood being labeled *Population* or *Therapy* of the ten most common labeled words in picoweb and ebmnlp. In contrast to the top 10 annotated words of the picoweb data set the top 10 annotated words of the ebmnlp data set only contain pharmaceutical ingredients. This showed us that ebmnlp annotations tend to be shorter and only contain the most essential elements.

Furthermore we put an emphasis on user feedback to get valuable insights into how to make the application as easy to use as possible without sacrificing too much annotation quality. This was achieved by implementing selected gamification techniques.

Next we used different variations of the picoweb data set as input for the SciBERT model to see how it performs. We also trained the SciBERT model on the ebmnlp data set to compare both performances.

As summarized in Table 4.13 including *sentences marked as not containing PICO elements* or PoS tags had almost no impact on the f1-score. All in all the SciBERT model achieved around 0.08 higher average f1-scores when trained on the variations of picoweb data set than on the ebmnlp data set.

| | **Pop-ula-tion** | **Inter-ven-tion** | **Outside** | **Average** |
|---|---|---|---|---|
| **picoweb excluding *sentences marked as not containing PICO elements*** | 0.86 | 0.83 | 0.92 | 0.87 |
| **picoweb including *sentences marked as not containing PICO elements*** | 0.87 | 0.83 | 0.92 | 0.87 |
| **picoweb w/o PoS tags** | 0.87 | 0.83 | 0.92 | 0.87 |
| **picoweb with PoS tags** | 0.87 | 0.84 | 0.92 | 0.88 |
| **ebmnlp** | 0.81 | 0.72 | 0.90 | 0.79 |
| **picoweb+ebmnlp** | 0.82 | 0.74 | 0.93 | 0.83 |

Table 4.13: Overview of f1-scores of experiments described in Section 4.3

CHAPTER 5

# Conclusion

We presented an analysis and comparison of two PICO data sets (Section 4.1). Even if both the picoweb (Section 2.1.1) and the ebmnlp (Section 2.1.2) data set consist of PICO labeled PubMed abstracts, the differences in the guidelines do have a high impact on the structure of the annotations. We presented experiments to show how the SciBERT model performs on these data sets (Section 4.3). The presented experiments describe the impact of different data set features (PoS, sentences without PICO elements, . . . ). We also presented ways to design an annotation platform as an eLearning system to enhance the motivation of the users.

## 5.1 eLearning platform

We created an eLearning system (Section 3.2) to provide an easy to use platform to input annotations of *population* and *therapy* elements in PubMed abstracts.

Additionally to the user feedback there are multiple topics that are worth having a deeper look in the future. We implemented the most promising gamification elements in our eLearning system, based on them it would be interesting to do additional experiment with more social elements like badges, levels, leagues, . . .

## 5.2 Machine Learning

Our first experiment presented in Section 4.3.1 answers the question what impact including sentences without PICO labels have on the performance of a ML model. It shows only a minimal improvement on the f1-score of population. This was not unexpected as the picoweb data set consists of 25820 sentences including PICO labels, but only 395 sentences without any PICO label. Redoing this experiment with more data could provide us with more insights on the impact of such sentences.

Our second experiment presented in Section 4.3.2 answers the question of what impact including PoS tags have on the performance of a ML model. These PoS tags were created using CoreNLP [Man+14]. This was done to explore the question if syntactic structure information lead to better performance. While the f1-scores did increase by 0.01, this had less impact than we expected. One can theorize that using a transformer like (Sci)Bert, the pre-training already learned similarities between words that would have the same PoS tag. Validating whether or not this can be the case would be a very interesting topic for future work.

To answer the question of how a model trained using the picoweb data set perform compared to the similar ebmnlp data set we trained the same model with each of the data sets. Comparing the performance of the ebmnlp data set (Section 4.3.3) to the picoweb (Section 4.3.1) shows a lower average f1-score by 0.08 and for intervention 0.11. As the ebmnlp guideline/annotation style differ from ours, one should be cautious to jump to conclusions too quickly. Reviewing manual samples of annotations in both data sets (Section 4.1) suggests that the ebmnlp annotations cover more diverse structured abstracts. On the other hand as most picoweb annotators do have at least basic knowledge of medicine and/or linguistics—while ebmnlp used Amazon Mechanical Turk annotators—this can also impact the quality of the data set.

Our last evaluation was on a combined data set (Section 4.3.4). This performed slightly better than the ebmnlp data set, but worse than the picoweb.

Looking on the differences found—between the picoweb and the ebmnlp data set—(Section 4.1), the results of the ML experiments in Section 4.3.4 we can answer the question if both data sets are compatible to combine to achieve better performance as follows. The differences in the annotation styles of these data sets prevent them to be simply merged to improve performance. This does not mean that this is impossible to achieve using more sophisticated methods, but that is out of the scope of this thesis.

An interesting topic for future work is using HMM-Crowd [Ngu+17] aggregation as done for the ebmnlp (Section 2.1.2) data set, as it probably could improve the achieved scores of the picoweb data set. Another interesting topic for future work is exploring ways to use the calculated reliability of a user to weight their annotations accordingly while training.

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| **Adam** | Adaptive Moment Estimation. 20, 21 |
| **AMT** | Amazon Mechanical Turk. 8, 58 |
| **BERT** | Bidirectional Encoder Representations from Transformers. 16, 17 |
| **BP** | backpropagation. 21 |
| **BRNN** | Bidirectional Recurrent Neural Network. 38 |
| **CBOW** | continuous bag of words. 15, 59 |
| **CBT** | clinical based trial. 1 |
| **CRF** | Conditional Random Field. 8 |
| **CS** | computer science. 17 |
| **DNN** | deep neural network. 21 |
| **EBM** | evidence-based medicine. 1 |
| **GD** | gradient descent. 18, 19 |
| **GloVe** | Global Vector. 14 |
| **GP** | general practitioner. 9 |
| **GPU** | graphics processing unit. 17 |
| **IAA** | inter-annotator agreement. 10, 11, 30, 49 |
| **ID** | identifier. 14 |
| **IMRD** | Introduction, Methods, Results, Discussion. 7 |
| **LSTM** | Long Short-Term Memory. 8, 22 |

| | |
|---|---|
| **MD** | medical doctor. 1, 7, 10 |
| **MeSH** | Medical Subject Headings. 28–30, 39, 49 |
| **ML** | Machine Learning. 2, 3, 5, 14, 17, 18, 20, 21, 23, 24, 39, 57, 58 |
| **NER** | Named Entity Recognition. 8 |
| **NLP** | Natural Language Processing. 7, 8, 16, 17 |
| **NNLM** | Feedforward Neural Net Language Model. 16 |
| **Pconf** | positive-confidence. 38 |
| **PICO** | Population, Interaction, Comparison, Outcome. xi, xiii, 2–4, 7, 8, 11, 32, 37–39, 48–52, 54–57, 60, 61 |
| **PoS** | Part of Speech. 3, 4, 8, 38, 39, 49, 52, 53, 55–58, 61 |
| **RCTs** | Randomized Control Trials. 7, 8 |
| **ReLU** | rectified linear unit. 18, 21, 22 |
| **RNN** | recurrent neural network. 22 |
| **SGD** | stochastic gradient descent. 6, 20 |

# Bibliography

[Arp+17] Devansh Arpit et al. „A closer look at memorization in deep networks“. In: *arXiv preprint arXiv:1706.05394* (2017).

[Ber10] Shane Bergsma. „Large-scale Semi-supervised Learning for Natural Language Processing“. AAINR67581. PhD thesis. Edmonton, Alta., Canada, 2010. ISBN: 978-0-494-67581-6.

[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)*. 2019, pp. 3606–3611.

[Bor+15] Antoine Bordes et al. „Large-scale simple question answering with memory networks“. In: *arXiv preprint arXiv:1506.02075* (2015).

[Bot10] Léon Bottou. „Large-scale machine learning with stochastic gradient descent“. In: *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[BSF94] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. „Learning long-term dependencies with gradient descent is difficult“. In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[Cai01] Roger Caillois. *Man, play, and games*. University of Illinois press, 2001.

[Chi+16] Billy Chiu et al. „How to train good word embeddings for biomedical NLP“. In: *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*. 2016, pp. 166–174.

[Coh60] Jacob Cohen. „A coefficient of agreement for nominal scales“. In: *Educational and psychological measurement* 20.1 (1960), pp. 37–46.

[Col+11] Ronan Collobert et al. „Natural language processing (almost) from scratch“. In: *Journal of Machine Learning Research* 12.Aug (2011), pp. 2493–2537.

[Cyb89] George Cybenko. „Approximation by superpositions of a sigmoidal function“. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[Det+11] Sebastian Deterding et al. „From game design elements to gamefulness: defining" gamification"". In: *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. 2011, pp. 9–15.

[Dev+18] Jacob Devlin et al. „Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[Dic+15] Darina Dicheva et al. „Gamification in Education: A Systematic Mapping Study". In: *Journal of Educational Technology & Society* 18.3 (2015), pp. 75–88. ISSN: 11763647, 14364522. URL: `http://www.jstor.org/stable/jeductechsoci.18.3.75`.

[DS79] Alexander Philip Dawid and Allan M Skene. „Maximum likelihood estimation of observer error-rates using the EM algorithm". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28.1 (1979), pp. 20–28.

[Erh+10] Dumitru Erhan et al. „Why does unsupervised pre-training help deep learning?" In: *Journal of Machine Learning Research* 11.Feb (2010), pp. 625–660.

[FES16] Alejandro Moreo Fernández, Andrea Esuli, and Fabrizio Sebastiani. „Distributional Correspondence Indexing for Cross-Lingual and Cross-Domain Sentiment Classification." In: *Journal of artificial intelligence research* 55 (2016), pp. 131–163.

[Fle71] Joseph L Fleiss. „Measuring nominal scale agreement among many raters." In: *Psychological bulletin* 76.5 (1971), p. 378.

[GBB11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. „Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 2011, pp. 315–323.

[Gla17] Tobias Glasmachers. „Limits of end-to-end learning". In: *arXiv preprint arXiv:1704.08305* (2017).

[Gob+14] Glenn T Gobbel et al. „Assisted annotation of medical free text using RapTAT". In: *Journal of the American Medical Informatics Association* 21.5 (2014), pp. 833–841.

[GS05] Alex Graves and Jürgen Schmidhuber. „Framewise phoneme classification with bidirectional LSTM and other neural network architectures". In: *Neural networks* 18.5-6 (2005), pp. 602–610.

[Gul+16] Caglar Gulcehre et al. „Pointing the unknown words". In: *arXiv preprint arXiv:1603.08148* (2016).

[HC18] David Hand and Peter Christen. „A note on using the F-measure for evaluating record linkage algorithms". In: *Statistics and Computing* 28.3 (2018), pp. 539–547.

[HM15] Julia Hirschberg and Christopher D Manning. „Advances in natural language processing“. In: *Science* 349.6245 (2015), pp. 261–266.

[Hoc91] Sepp Hochreiter. „Untersuchungen zu dynamischen neuronalen Netzen“. In: *Diploma, Technische Universität München* 91.1 (1991).

[Hoc98] Sepp Hochreiter. „The vanishing gradient problem during learning recurrent neural nets and problem solutions“. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), pp. 107–116.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. „Long short-term memory“. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[Hua98] Zhexue Huang. „Extensions to the k-means algorithm for clustering large data sets with categorical values“. In: *Data mining and knowledge discovery* 2.3 (1998), pp. 283–304.

[INS18] Takashi Ishida, Gang Niu, and Masashi Sugiyama. „Binary Classification from Positive-Confidence Data“. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 5917–5928. URL: `http://papers.nips.cc/paper/7832-binary-classification-from-positive-confidence-data.pdf`.

[JR71] Nick Jardine and Cornelis Joost van Rijsbergen. „The use of hierarchic clustering in information retrieval“. In: *Information storage and retrieval* 7.5 (1971), pp. 217–240.

[KB14] Diederik P Kingma and Jimmy Ba. „Adam: A method for stochastic optimization“. In: *arXiv preprint arXiv:1412.6980* (2014).

[KSB14] Rafał Kurczab, Sabina Smusz, and Andrzej J Bojarski. „The influence of negative training set size on machine learning-based virtual screening“. In: *Journal of cheminformatics* 6.1 (2014), p. 32.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. „Imagenet classification with deep convolutional neural networks“. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[KZW19] Tian Kang, Shirui Zou, and Chunhua Weng. „Pretraining to recognize piCO elements from randomized controlled trial literature“. In: *Studies in health technology and informatics* 264 (2019), p. 188.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. „Deep learning“. In: *nature* 521.7553 (2015), p. 436.

[LH13] Baoli Li and Liping Han. „Distance weighted cosine similarity measure for text classification“. In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2013, pp. 611–618.

[LH17] Ilya Loshchilov and Frank Hutter. „Decoupled weight decay regularization“. In: *arXiv preprint arXiv:1711.05101* (2017).

[Man+14] Christopher Manning et al. „The Stanford CoreNLP natural language processing toolkit“. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 2014, pp. 55–60.

[Mer01] Sharan B Merriam. „Andragogy and self-directed learning: Pillars of adult learning theory“. In: *New directions for adult and continuing education* 2001.89 (2001), pp. 3–14.

[Mik+13] Tomas Mikolov et al. „Efficient estimation of word representations in vector space“. In: *arXiv preprint arXiv:1301.3781* (2013).

[MYZ13] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. „Linguistic regularities in continuous space word representations“. In: *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*. 2013, pp. 746–751.

[Ngu+17] An T Nguyen et al. „Aggregating and predicting sequence labels from crowd annotations“. In: *Proceedings of the conference. Association for Computational Linguistics. Meeting*. Vol. 2017. NIH Public Access. 2017, p. 299.

[Nwo97] Kevin Ngozi Nwogu. „The medical research paper: Structure and functions“. In: *English for specific purposes* 16.2 (1997), pp. 119–138.

[Nye+18] Benjamin Nye et al. „A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature“. In: *Proceedings of the conference. Association for Computational Linguistics. Meeting*. Vol. 2018. NIH Public Access. 2018, p. 197.

[Pak+10] Serguei Pakhomov et al. „Semantic similarity and relatedness between clinical terms: an experimental study“. In: *AMIA annual symposium proceedings*. Vol. 2010. American Medical Informatics Association. 2010, p. 572.

[Pas+19] Adam Paszke et al. „PyTorch: An Imperative Style, High-Performance Deep Learning Library“. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[Pet82] Catherine Pettinari. „The function of a grammatical alternation in 14 surgical reports“. In: *Linguistics and Literacy*. Springer, 1982, pp. 145–185.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. „Glove: Global vectors for word representation“. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[Qia99] Ning Qian. „On the momentum term in gradient descent learning algorithms“. In: *Neural networks* 12.1 (1999), pp. 145–151.

[RD95] William Rosenberg and Anna Donald. „Evidence based medicine: an approach to clinical problem-solving.“ In: *BMJ: British Medical Journal* 310.6987 (1995), p. 1122.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. „Learning representations by back-propagating errors“. In: *nature* 323.6088 (1986), p. 533.

[Ric+95] W Scott Richardson et al. „The well-built clinical question: a key to evidence-based decisions“. In: *ACP journal club* 123.3 (1995), A12–A12.

[RML06] Jorge G Ruiz, Michael J Mintzer, and Rosanne M Leipzig. „The impact of e-learning in medical education“. In: *Academic medicine* 81.3 (2006), pp. 207–212.

[Ros58] Frank Rosenblatt. „The perceptron: a probabilistic model for information storage and organization in the brain.“ In: *Psychological review* 65.6 (1958), p. 386.

[Rud16] Sebastian Ruder. „An overview of gradient descent optimization algorithms“. In: *arXiv preprint arXiv:1609.04747* (2016).

[Sch+07] Connie Schardt et al. „Utilization of the PICO framework to improve searching PubMed for clinical questions“. In: *BMC medical informatics and decision making* 7.1 (2007), p. 16.

[Sco55] William A Scott. „Reliability of content analysis: The case of nominal scale coding“. In: *Public opinion quarterly* (1955), pp. 321–325.

[SD03] Erik F Sang and Fien De Meulder. „Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition“. In: *arXiv preprint cs/0306050* (2003).

[Seo+16] Miran Seok et al. „Named entity recognition using word embedding as a feature“. In: *Int. J. Softw. Eng. Appl* 10.2 (2016), pp. 93–104.

[SF07] Yutaka Sasaki and R Fellow. „The truth of the F-measure, Manchester: MIB-School of Computer Science“. In: *University of Manchester* (2007).

[Sha48] Claude E Shannon. „A mathematical theory of communication“. In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[Sno+08] Rion Snow et al. „Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks“. In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics. 2008, pp. 254–263.

[SP97] Mike Schuster and Kuldip K Paliwal. „Bidirectional recurrent neural networks“. In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[Sri+14] Nitish Srivastava et al. „Dropout: a simple way to prevent neural networks from overfitting“. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[TSA15] MUNEEB TH, Sunil Sahu, and Ashish Anand. „Evaluating distributed word representations for capturing semantics of biomedical concepts“. In: *Proceedings of BioNLP 15* (2015), pp. 158–163.

[Zhu+15] Yukun Zhu et al. „Aligning books and movies: Towards story-like visual explanations by watching movies and reading books“. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 19–27.

[Zla+18a] Markus Zlabinger et al. „Extracting the Population, Intervention, Comparison and Sentiment from Randomized Controlled Trials“. In: *Building Continents of Knowledge in Oceans of Data: The Future of Co-Created eHealth - Proceedings of MIE 2018, Medical Informatics Europe, Gothenburg, Sweden, April 24-26, 2018*. 2018, pp. 146–150. DOI: `10.3233/978-1-61499-852-5-146`. URL: `https://doi.org/10.3233/978-1-61499-852-5-146`.

[Zla+18b] Markus Zlabinger et al. „Medical Entity Corpus with PICO elements and Sentiment Analysis“. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Ed. by Nicoletta Calzolari (Conference chair) et al. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. ISBN: 979-10-95546-00-9.