Informatics

Text analysis using colexification networks

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Armin Gander, BSc
Matrikelnummer 11848230

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ. Prof. Dr. Allan Hanbury
Mitwirkung: Univ.-Prof. Dr. David Garcia
M.Sc. Anna Di Natale

Wien, 20. Mai 2021

Armin Gander

Allan Hanbury

Text analysis using colexification networks

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Armin Gander, BSc
Registration Number 11848230

to the Faculty of Informatics

at the TU Wien

Advisor: Univ. Prof. Dr. Allan Hanbury
Assistance: Univ.-Prof. Dr. David Garcia
M.Sc. Anna Di Natale

Vienna, 20th May, 2021

Armin Gander   Allan Hanbury

Erklärung zur Verfassung der Arbeit

Armin Gander, BSc
Schottenfeldgasse 86/7, 1070 Wien

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, 20. Mai 2021

Armin Gander

Danksagung

Ich möchte mich an dieser Stelle bei allen Menschen bedanken, die mich bei der Arbeit an dieser Masterarbeit unterstützt haben.

Einen sehr wichtigen Beitrag leisteten M.Sc. Anna Di Natale und Univ.-Prof. Dr. David Garcia, die mich sehr aufmerksam und stets hilfsbereit betreut haben. Durch wöchentliche gemeinsame Treffen und ihre kontinuierliche Unterstützung haben sie mir eine perfekte Möglichkeiten geschaffen, mich bei der Forschung des Complexity Science Hub Vienna einzubringen und meinen Beitrag zu leisten.

Zudem möchte mich bei allen Mitwirkenden des Vienna Scientific Cluster (VSC) für den kostenlosen Zugang zum leistungsstärksten Supercomputer Österreichs bedanken. Ohne diese hervorragende Ressource für Rechenleistung miteingenommen der hervorragend durch Dipl.-Ing. Dr. Claudia Blaas-Schenner geleiteten VSC-Einführungskursen wäre diese Arbeit nicht möglich gewesen.

Dieses Forschungsprojekt wurde vom Wiener Wissenschafts-, Forschungs- und Technologiefonds durch das Projekt 'Emotionales Wohlbefinden in der digitalen Gesellschaft' (Förderungsnummer VRG16-005) gefördert, welches an Prof. David Garcia vergeben wurde. Dies ermöglichte den Zugang zum Vienna Scientific Cluster (VSC) und dem Corpus of Historical American English (COHA), welche sehr wertvolle Ressourcen für diese Forschung waren.

Zuletzt möchte ich noch meinen Eltern, meinen Brüdern, meiner Schwester und allen meinen Freunden dafür bedanken, dass sie mir immer zur Seite stehen und gerne Zeit mit mir verbringen.

Acknowledgements

I would like to take this opportunity to thank all the people who have supported me in my work on this master's thesis.

A very important contribution was made by M.Sc. Anna Di Natale and Univ.-Prof. Dr. David Garcia, who have been very attentive and always helpful in supervising me. Through weekly joint meetings and their continuous support, they created the perfect opportunities for me to get involved and contribute to the research of the Complexity Science Hub Vienna.

In addition, I would like to thank all the contributors of the Vienna Scientific Cluster (VSC) for the free access to the most powerful supercomputer in Austria. This work would not have been possible without this excellent resource for computational power along with the excellent VSC introductory courses taught by Dipl.-Ing. Dr. Claudia Blaas-Schenner.

This research project received funding from the Vienna Science and Technology Fund through the project "Emotional Well-Being in the Digital Society" (Grant No. VRG16-005) awarded to Prof. David Garcia. This granted access to the Vienna Scientific Cluster (VSC) and the Corpus of Historical American English (COHA), which have been very valuable resources for this research.

Last but not least, I would like to thank my parents, my brothers, my sister, and all my friends for supporting and spending time with me.

Kurzfassung

Das Phänomen der Kolexifikation beschreibt Vorkommnisse in der natürlichen Sprache, bei denen zwei Konzepte durch das gleiche Wort in mindestens einer Sprache ausgedrückt werden. Wir nutzen dieses linguistische Prinzip, um eine theoriegeleitete Textanalysemethode zu konstruieren. Im Vergleich zu vielen State-of-the-Art-Modellen für die Verarbeitung natürlicher Sprache (NLP) ist diese Methode vollständig interpretierbar und erlaubt präzise Einblicke in die Struktur des Modells. Solche theoriegeleiteten Ansätze sind zunehmend gefragt, da es bei vielen großen NLP-Modellen für Entwickler schwierig ist, die Dynamik der Modelle und deren Implikationen zu verstehen.

Die hier vorgeschlagene Textanalysemethode basiert auf einem Wortähnlichkeitsmaß, das auf einem Kolexifikationsnetzwerk aufgebaut ist, d.h. einem Netzwerk von Konzepten, die durch das Auftreten von Kolexifikationen verbunden sind. Inspiriert von ähnlichen Ansätzen in anderen Domänen, berechnen wir das Wortähnlichkeitsmaß als stationäre Besuchsverteilung in jedem Knoten und validieren es mit mehreren der meistverwendeten Wortähnlichkeitsdatensätze in NLP. Die Ergebnisse der Validierung anhand von Wordähnlichkeitsdatensätzen zeigen, dass die auf Kolexifikation basierende Methode vergleichbare Methoden deutlich übertrifft. Nach der Validierung der Wortähnlichkeitsmetrik definieren wir ein Textähnlichkeitsmaß. In verschiedenen Experimenten, die auf Datenbanken mit englischen Texten basieren, validieren wir das Maß, indem wir zeigen, dass es in der Lage ist, Texte auf der Grundlage ihres Genres, Autors und ihrer Herkunft mit angemessener Genauigkeit zu unterscheiden. Wir vergleichen die Ergebnisse der Methode mit denen eines Standard-Ansatzes zur Textanalyse und stellen fest, dass die beiden Modelle zu vergleichbaren Ergebnissen führen.

Die in dieser Arbeit entwickelte Textanalysemethode erlaubt es uns, die Hypothese zu validieren, dass Kolexifikationsvorkommen semantische Beziehungen zwischen Konzepten kodieren, und zu zeigen, dass ein auf Kolexifikation basierender Ansatz in verschiedenen Textanalyseaufgaben signifikante Vorzüge hat und zu sinnvollen Erkenntnissen führt. So führen wir beispielsweise eine historische Analyse amerikanisch-englischer Belletristik durch und zeigen, dass Stil und Inhalt der Belletristik im Laufe der Zeit vielfältiger geworden sind. Vor allem in den letzten Jahrzehnten hat die Rate der Veränderung stark zugenommen. Diese Erkenntnisse stimmen überein mit anderen Erkenntnissen aus der computergestützten Sozialwissenschaft, welche darauf hindeuten, dass der Fluss kultureller Inhalte in den letzten Jahrzehnten zugenommen hat.

Abstract

The phenomenon of colexification describes occurrences in natural language in which two concepts are expressed by the same word in at least one language. We deploy this linguistic principle to construct a theory-driven text analysis method. Compared to many state-of-the-art natural language processing (NLP) models, this method is fully interpretable, allowing precise insights into the structure of the model. Such theory-driven approaches are increasingly in demand since when using other large NLP models it is difficult for developers to understand a models' dynamics and implications thereof. Furthermore, the proposed method is domain-independent because it is constructed on the language-layer itself as compared to the majority of state-of-the-art methods, which are trained using large corpora of texts.

The text analysis method here proposed is based on a word similarity measure built on top of a colexification network, i.e. a network of concepts linked by occurrences of colexification. Inspired by similar approaches in other domains, we compute the word similarity measure as the stationary visiting distribution in each node and validate it using several of the most used word similarity datasets in NLP. The results show that the colexification-based method significantly outperforms other word and graph embedding approaches in the task of word similarity prediction. After the validation of the word similarity metric we define a text similarity measure inspired by a state-of-the-art approach to the same task. Performing various experiments based on databases of English texts, we validate the measure by showing that it is able to distinguish text excerpts on the basis of their genre, author and text of origin with reasonable accuracy. We compare the results of the method with the ones of a standard NLP approach on the genre recognition task and find that the two models reach comparable performances.

The text analysis method developed in this work allows us to validate the hypothesis that colexification occurrences encode semantic relationships between concepts. Furthermore, we show that a colexification-based approach to NLP has significant merits in various text analysis tasks, leading to meaningful insights. For instance, we perform a historical analysis of American English fiction literature, showing that the style and content of fiction literature has become more diverse over time, with the rate of change increasing particularly sharply in recent decades. These insights can be linked to other findings in computational social science, suggesting that the flux of cultural content has been increasing during the last decades.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Natural Language Processing (NLP) constitutes one of the frontiers of machine learning. Even before the introduction of machine learning techniques, the dream of building a machine capable of interacting with humans has characterized the technological research since the invention of computers. At the times, NLP tasks were deemed achievable by machines, and the development of a system so intelligent to be indistinguishable from humans was a tangible concern. This fear inspired Turing to develop the imitation game, also known as Turing test, in 1950 [Tur50]. Such test was aimed at assessing whether machines could mimic humans so well to be indistinguishable from them. The Turing test was based on a written conversation, which shows how the idea of human-like intelligence is intrinsically related to the ability of understanding language and producing answers in the same linguistic system. At the time, experts thought that it would have taken a few years to produce a machine able to translate language and engage in conversation. However, the understanding of human language has proven to be an hard task, which to date only humans can master. Moreover, NLP problems have retained their relevance in the field and they are one of the main branches of machine learning and artificial intelligence research.

The first approaches to the understanding and analysis of language were based on a set of linguistically accepted rules. However, in the chase for better performance, the theoretical validity and the explainability of the systems was lost. Over time, theory-driven models have been substituted by corpus-based methods and the quest for a better understanding of human language has been replaced by the race for a method that outperform the previous ones only by a few decimals. Some systems are claimed to reach performances that are better than humans [HG14], even if the meaning of such declamations is not clear. At the moment, state-of-the-art tools are based on knowledge retrieved from big corpora of texts and on black box models. These models reach high performance, but due to the lack of explainability it is currently impossible to comprehend how exactly the results are achieved. Thus, such models are very difficult to interpret and audit. Indeed, these methods allow the analysis of language but hardly enable the extraction of knowledge. In detail, these tools perform well in NLP tasks, but their performance is not related to a set of knowledge we can subsequently apply to other problems. In other words, the lack in explainability of the results poses a great challenge in moving the field towards a more complete understanding of human natural language, nor does it allow the researcher to investigate the reasons of the outcomes.

This trend of ever increasing, non-interpretable NLP models comes with several flaws and risks, as has been shown in [BGMMS21]. On one hand, the lack of explainability makes it impossible for the users and developers to adapt the system to their realm of analysis and to control the hidden biases of a model and thus reinforcing a hegemonic worldview. It has been shown, for example, that language models such as BERT [DCLT18], one of the most prominent model developed in recent years, rely almost entirely on "exploitation of spurious statistical cues in the dataset" [NK19] without actually understand the syntax or semantics of a text. Due to the increasing sizes of the text corpora necessary for training the models in question, it is impossible to manually curate and document the training data. Thus, "the ability of language models to pick up on both subtle biases and overtly abusive language patterns in training data leads to risks of harms" [BGMMS21]. Such problems are accentuated by the fact that large, Internet-based datasets, which are used predominantly for training, tend to neglect the worldview of people at the margins of our society. Furthermore, large NLP models which were trained on "petabytes of data collected over 8 years of web crawling" [BCjC19] are rarely flexible enough to adapt to changing social views in society, which clearly reinforced the hegemonic interpretation of the world.

Furthermore, the development of large language models such as BERT [DCLT18] comes with significant environmental risks due to their energy consumption as well as opportunity cost of redirecting research efforts and resources, which could be used for developing language models that actually capture the meaning of texts instead of exploiting statistical properties thereof [BGMMS21]. While the training of big NLP models does indeed consume great amounts of energy, as analyzed in [SGM19], it can also be argued that the increasing interest and importance of NLP models in daily life justify such an increase in used resources. For instance, NLP models are increasingly used in many daily life applications, including domains such as the health sector [HWSU20], [SV21] or in commerce [XRK$^+$21]. Furthermore, the increasing relevance of NLP in public consciousness as well as in the academic field can be confirmed by inspecting the frequency of appearances of the term 'natural language processing' in literary sources. Figure 1.1 presents the relative frequency of appearance of the term 'natural language processing' from the year 1500 to 2019 retrieved from Google Books Ngram Viewer[1]. We can observe

[1] https://books.google.com/ngrams

two steep increases in the popularity of the term, one beginning in the 1960s and one in the late 2000s.

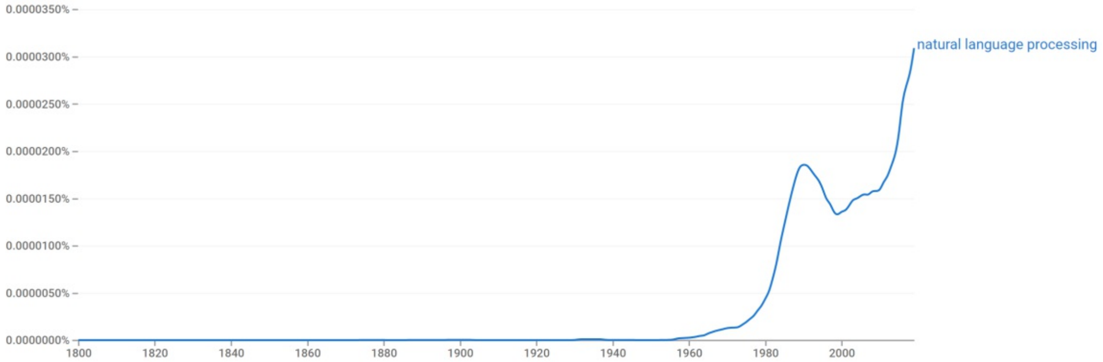


Figure 1.1: Relative frequency of the term 'natural language processing' in literary sources published between 1500 and 2019 from Google Books Ngram Viewer

Deviating our work from the ant race of the late machine learning quest for better performance, we deem explainable systems and theory-driven approaches more valuable than empty black box systems. Indeed, we believe that the knowledge profit of such approaches can fully repay the possible loss in performance of the system. In this thesis, we build a theory-based method for NLP and explore the possibilities of application of such tool. In particular, the tool will deploy on one aspect of language that represent a limitation for state-of-the-art machine learning approaches, i.e. ambiguous words or, more precisely, colexifications. We base our method on the properties of such phenomenon and the reasons of its appearance in language.

1.2 Problem statement

In natural language, some words denote more than one concept. For example, the word 'stock' can be used to talk about a financial product as well as a liquid used as a basis for soup. This phenomenon, namely that a single word form can have multiple meanings in human language, poses a fundamental problem in the field that deals with the computational analysis of language, i.e Natural Language Processing (NLP). Ambiguous words represent a significant challenge for computational models, since explicit distinction rules between multiple meanings of a word are hard to define. The interpretation of ambiguous words still shows a significant gap between a computer's insight and the human understanding of natural language. Indeed, one of the latest benchmark for language understanding systems, SuperGLUE [WPN$^+$19], includes a task to assess such ability. In particular, SuperGLUE contains the Word in Context task (WiC), where a polysemous word is displayed in two different contexts and the system has to assess whether it is used with the same sense in both contexts. State-of-the-art approaches, like SenseBERT-large [LLD$^+$20], have still room for improvement on this task when put in comparison to the human performance. Indeed, understanding ambiguous words and sentences requires a level of verbal intelligence and semantic knowledge which is unusual for a computer. In this work, we propose a method that deploys the problem of ambiguous words as a resource. More in detail, we construct a knowledge- and theory-based text analysis method which deploys the occurrences of ambiguous words to acquire insights into the structure of language.

So far, methods of text representation and analysis can be divided into three main categories. Corpus-based models are grounded on the analysis of large text corpora and on the assumption that the semantic meaning of a word can be inferred from its position inside the structure of sentences and texts. Knowledge-based methods make use of semantic networks, i.e. databases which include explicit ontological relationships between words, as for example WordNet ([MBF+90]). A third category, which includes features from the previous two, has been established with the introduction of hybrid systems. Here, we present a method that falls into the second category and which could be used, in the future, as basis for an hybrid system.

As previously mentioned, our work is based on occurrences of ambiguous words, which in linguistics are called colexifications. The concept of colexification describes the phenomenon in which two different meanings are expressed using the same word in one language. By comparing colexifications and their characteristic distribution, linguists gain insights into a wide variety of aspects, including human perception [JWH+19] and the evolution of language. Most of this research is conducted with the use of colexification networks, which are built from databases of occurrences of such phenomenon. The current applications of colexification networks mostly rely on the hypothesis that colexification occurrences hint to some kind of meaning similarity. This idea has been first hypothesized by François in the work that defined the idea of colexification [Fra08], and deployed hereafter, but it still lacks validation at scale. Recently, in [DNPG21] such validation has been performed in the realm of affective meaning, but a complete test of such hypothesis is still needed. The phenomenon of colexification and colexification networks are a relatively new research topics, still in need of further investigation. Moreover, the interest toward colexification has been increasing recently, especially in fields different from linguistics. Indeed, this concept has promising features that can lead to novel approaches in interdisciplinary research. In particular, in this thesis we will explore the possibility of deploying colexification structures in the area of NLP.

The central hypothesis of our work resides in the fact that colexification occurrences encode semantic meaning. From this idea follows the construction of a tool for the analysis of texts based on colexification databases. If this tool revealed to be successful, we would not only prove that linguistic theory can be employed for computer science tasks in a productive way, but also that the founding hypothesis of this project is reasonable. However, the analysis here reported cannot result in a theoretical validation of said hypothesis, which needs different efforts to be confirmed.

Natural Language Processing deals with the construction of tools for the analysis of texts in human language. A central task in NLP is the comparison of different corpora of texts according to their structure and content, as for example in authorship attribution and text similarity tasks. In order to enable this comparison deploying colexification networks, we have to define an underlying measure for the similarity of nodes. Subsequently, we have to consider an aggregation algorithm that converts the word similarity into a similarity score for entire texts. In this thesis, we consider different ways of transforming such similarity into a measure that operates on the level of entire texts. The employment of colexification networks for the computation of word similarity allows taking into account the variation of conceptual understanding of different languages. Indeed, it is known that different cultures interpret some concepts, for example emotions [JWH$^+$19], in different ways. Thus, the text analysis method resulting from this project will encompass a high diversity of cultural interpretations of our world and will aim at being universally valid.

State-of-the-art NLP methods show some limitations. Firstly, it is known that their results lack explainability. This is a big concern when deploying non-interpretable models, as Latent Semantic Analysis (LSA) [DDF$^+$90] or Latent Dirichlet Allocation (LDA) [BNJ03], because they do not allow for a reconstruction of the process that brought to a certain output. The impossibility of understanding which feature of the text is used to produce a result is a concerning problem. Indeed, it has been shown that some of these systems have adopted biases from human texts and use them to produce their outputs [RH21]. Moreover, some of these algorithm are based on assumptions that have little theoretical significance. Therefore, the outputs of such models seldom offer valuable theoretical insights into the structure of natural language. In addition to this, the reliance of corpus-based approaches on a text corpus makes these methods inherently domain specific. The method proposed here aims at alleviating these drawbacks by relying on a theoretical concept, colexification. This ensures the consideration of semantic relations between terms as well as the adaptability to most domains of textual expression. Moreover, its transparency will allow for the acquisition of further insights into the structure of language. In order to allow the understanding and reproduction of the results of this work, we published a GitHub repository under the Apache 2.0 licence, which can be found here[2].

1.3 Objective

The aim of this work is to build a theory-driven text analysis method able to provide theoretical insight into the structure of natural language. Such a method is based on the linguistic concept of colexification. As the resulting text analysis tool is theory-driven, it allows a linguistic interpretation of the results. On the contrary, this is not possible with many state-of-the-art text analysis methods, which rely mostly on black box models. Therefore, we propose a purely knowledge-based approach to text analysis, in order to ensure maximum interpretablity. Furthermore, this study pursues the goal of proving that colexification networks can be applied to NLP problems, which can lead to novel results and interesting insights into the properties of natural language. Moreover, the

[2] https://github.com/gander-a/Output

interpretability of the method allows for more informed changes and modifications. In this thesis, first we design a similarity metric that satisfies a set of predefined axioms. Secondly, we implement such similarity metric and build a text analysis tool that deploys it. Such tool will take two texts as input and will compute the resulting text similarity. The validity of the text analysis measure is then tested on a set of empirical scenarios. Finally, we will conduct some experimental analyses to examine other areas of applications of our algorithm.

1.4 Outline

In **chapter 2**, we explore the scientific background which is needed for the understanding of this work. We review the most important methods in the fields of text similarity, text analysis and NLP. We also introduce the concept of colexification and report on its previous interdisciplinary applications.

In **chapter 3**, we present the methods put in place for the construction of the text analysis method. We first define a distance measure between words in the network and we consider different ways of aggregating it into a text-level measure. Moreover, we present the setting for the experiments and the tools used to quantify the results.

In **chapter 4**, we report the experiments conducted in the scope of this work. In particular, we divide them into validation experiments, which give us insights into the performance of the method, and exploration experiments, which have the aim of exploring the potentialities of the method.

In **chapter 5**, we report the results of the two categories of experiments and we discuss the outcomes.

Finally, **chapter 6** includes a summary of the findings of this study and an outlook for future work.

CHAPTER 2

Background

In this chapter, we review the background information for our work. We will start by introducing the linguistic concept of colexification and report on its applications to interdisciplinary problems in section 2.1. Afterwards, we review the most relevant text similarity algorithms in section 2.2.

2.1 Colexification

The term colexification describes cross-linguistic lexical association patterns. This phenomenon, originally defined in linguistics, occurs when one word is used to express multiple concepts. When two or more meanings are expressed using the same word in a language, such language is said to colexify the two meanings [Fra08]. For example, the word 'Tau' in German denotes multiple different concepts: water condensed due to a temperature difference, a twisted rope and the 19th letter of the Greek alphabet. Thus, German is said to colexify those three concepts. The actual meaning of the German word 'Tau', therefore, is ambiguous and must be inferred from the context. In general, lexical ambiguity can be divided into homonymy and polysemy, depending on whether or not the meanings are related. In particular, in the case of polysemy, the meanings that a word conveys are in some way related, while for homonymy they are not. From a different perspective, we can describe polysemy as the phenomenon that a word may have more than one meaning and homonymy as the phenomenon that two or more concepts share the same word form [Pan82].

More in detail, homonymous words share the same word form and completely different and semantically independent meanings. One example of homonyms is the word 'bear', which indicates both an animal, and, as a verb, conveys the meaning of 'enduring'. In general, words that are historically derived from distinct lexical items are considered to be homonymous [Kle02], i.e. homonomy is observed in lexical items that carry two distinct and unrelated meanings and accidentally converge to the same word form [Kro97]. The context and the discourse setting help in their disambiguation [WW64]. In linguistics, the term homonymy encompasses both the concepts of homophony and homography. Homophony defines similar-sounding words with completely different meanings and different linguistic histories. For example, the two words 'heir' and 'air' have a similar pronunciation but different word forms and meanings. On the contrary, the concept of homography indicates words that share the same spelling but have different meanings. An example for this is the English word 'well', which can indicate 'good health' as well as an 'underground water source'.

On the opposite, polysemy describes cases in natural language when multiple related meanings are described by one and the same expression. In other words, polysemy denotes cases of single words having two or more related senses [Tar09], which constitute a partial representation of the overall concept [Kro97]. In most cases, the ambiguity consists in the fact that the meanings of two or more words overlap in one area and can be described as synonymous in this respect, but are not interchangeable in other areas. Occasionally, the term 'conceptual overlap' is also used to describe this phenomenon [Eri17]. Also in this case, the respective partial meaning can only be recognised from the context. An example of polysemy can be found in the word 'healthy', which means 'physically fit' (e.g. *The boy is healthy*) as well as 'supportive of health' (e.g. *The meal is healthy*) [FV15].

However, a clear distinction between homonomy and polysemy does not exist. The idea of relatedness of concepts is labile and no formal identity between lexical items can be given since meaning and meaning similarity are not binary values but belong to a spectrum [Pan82]. Thus, an exact demarcation line between homonymy and polysemy can not be established [Tar09]. Moreover, some words can be both polysemous and homonyms. For example, the word 'court' can be interpreted semantically as 'dish', 'meal' or 'court building'. Furthermore, it is connected with technical meanings from the legal language such as 'court' (as an institution) or 'judicial body' (committee). Therefore, the word 'court' can be seen as a polysemy when taking into account the meanings 'dish' and 'meal' but as an homonymy when considering the third meaning, the one connected to the judicial body. In this example, there is complex ambiguity since the word can be used in both, a homographic as well as a polysemous sense [Eri17]. Examples such as this demonstrate that the concepts of homonomy and polysemy are not mutually exclusive, but they are two aspects of the same phenomenon. Thus, François [Fra08] proposes to replace the two interpretative terms homonomy and polysemy with the descriptive term 'colexification'. In detail, a language "is said to colexify two functionally distinct senses if, and only if, it can associate them with the same lexical form" [Fra08]. The origination of colexifications can be manifold: while some patterns of colexification are based on a similarity in meaning, others can arise because of geographical, historical reasons or out of mere coincidence. An illustration showing these phenomena using the example of the word 'court' is depicted in Figure 2.1.

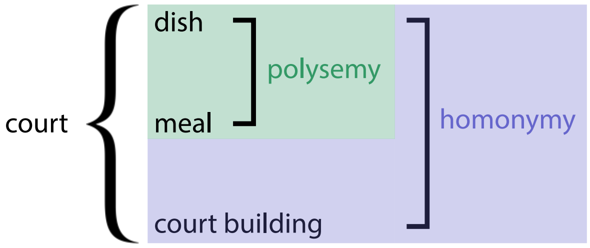


Figure 2.1: The English word 'court' is an example of word whose senses are both polysemic and homonymic.

2.1.1 Colexification networks

Databases of colexification occurrences across languages can be used to construct colexification networks. In colexification networks, concepts are connected if they are colexified by the same word in at least one language. As an example, in the language Manchu the same word ('$suk^h tun$') is used to describe the concepts 'air' and 'breath'. Thus, the concepts 'air' and 'breath' are connected in the colexification network. Figure 2.2 depicts the process of construction of a colexification network. Following the idea of François [Fra08], the underlying assumption is that the links in colexification networks follows the semantic similarity between concepts or words. In colexification networks, edges can be weighted according to the number of languages by which the two concepts are colexified. Furthermore, languages can be categorized into language families and link weights in colexification networks can alternatively be defined as the number of language families that present the same colexification pattern. In order to remove non-informative edges in the colexification networks, i.e. colexification occurrences due to historical processes, social and geographical environments, coincidence or errors, and to keep only links related to semantic associations, a subset of the original network is traditionally considered. In particular, a threshold for the number of languages and the number of families (i.e. the edge weight) is set and only links with weights above the thresholds are considered.

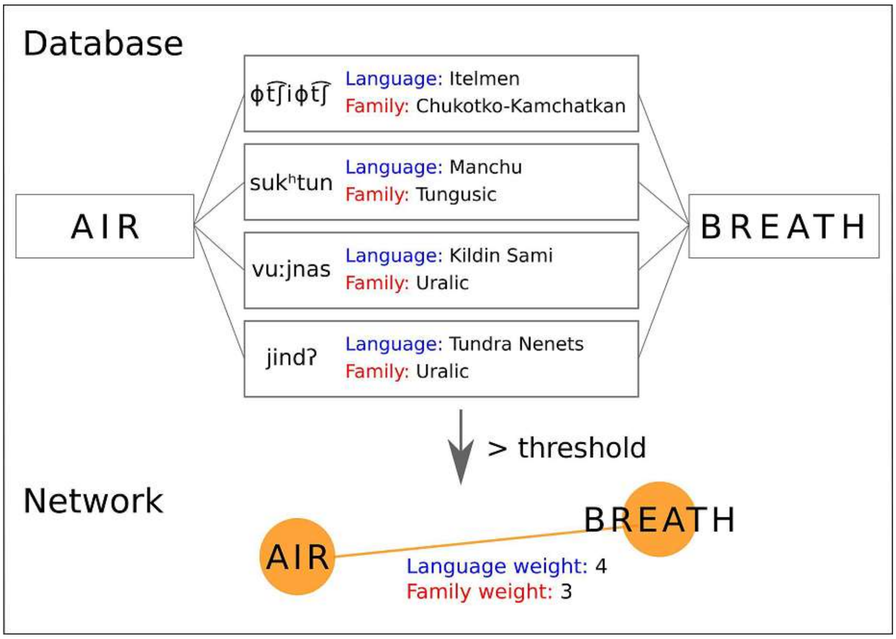


Figure 2.2: Colexification instances across languages can be collected in a database. Here is reported the example of the word pair 'air' and 'breath'. From the database it is possible to build a colexification network considering only colexification patterns with weights higher than a threshold [DNPG21].

Since the construction of colexification databases requires expert knowledge in linguistics, such databases have to be compiled manually. Indeed, the idea of concept is labile and hard to define. It is however possible to approximate colexification occurrences using multilingual dictionaries, collecting instances of identical translations [DNPG21]. Identical translations occur when two different words in one language are translated into the same word in another language. While the phenomenon of colexification is based on the notion of concepts, identical translations deal with words. For example, considering the word 'back', it is possible to distinguish between the sense indicating a direction and the meaning that refers to a part of the human body. Since these are two different concepts, they are distinguished in a colexification database. On the opposite, if the two senses were expressed by the same word in all the languages considered, then this pattern would not be included in the identical translation database. Thus, the amount of colexification patterns that can be recovered using identical translation is variable and depends on the language coverage of the database. However, it is possible that some colexifications might never appear in a database of identical translations, despite the number of languages it analyses. In short, in some cases identical translations do not recover colexifications and the percentage of recovered patterns depends on the number of languages that the database features. Therefore, identical translations are considered as an approximation of colexifications. Colexification networks are more sophisticated as they distinguish between two concepts of polysemous words whereas networks based on words, inherently represent one distinct word form as a single node in the network. The main advantage of identical translations is the possibility to automatically detect them from bilingual dictionaries. Thus, they can incorporate much more data and increase their scope substantially. Once identical translation patterns are stored in a databases, it is possible to build networks where words are connected by a link if they can be identically translated in at least one language [DNPG21]. These networks are also called colexification networks. Due to the symmetric nature of translations the networks are undirected. Similarly to [LGA$^+$18], the links between the network's nodes are associated with a weight corresponding to the number of languages (alternatively, the number of families of languages) which encompass the respective lexical relationship.

2.1.2 Databases

In this work, we consider colexification networks constructed from three different databases: one colexification database, Clics^3, and two identical translation databases, OmegaWiki and FreeDict. Below we present these databases in more detail:

- **Clics^3** [RTG$^+$20]
  The Clics^3 database (Database of Cross-Linguistic Colexifications) is a structured collection of colexification occurrences and is one of the most extensive linguistic resources considering language coverage. This database and its previous versions have been used to analyse colexification patterns in various fields beyond linguistics [DNPG21]. The network encompasses 3,156 language varieties coming from 30 different language databases and collects 4,228 different colexification patterns. It is based on the definitions and classifications of concepts of Concepticon [LGA$^+$18] and it is the world's largest existing database of comparative language colexifications. As the underlying databases were annotated manually by experts, the quality of data it features is very high. However, its size is not optimal for the development of text analysis methods, since it contains few words compared to other databases. Furthermore, the style of concept definitions translates into difficulties of application to natural language. For example, concepts are disambiguated adding parenthetical information (e.g. 'wash (oneself)' vs 'wash (clothes)'), which is challenging to use when automatically processing big amount of texts. In fact, some text analysis tools are based on direct string-comparison of single words and expressions, which are not designed for taking into account the additional information. On the contrary, some ambiguous words are included in the database without disambiguation. For example, the node corresponding to the concept 'head' is linked to concepts like 'boss' and 'chieftain', as well as to 'brain' and 'skull'. Therefore, we can infer that the node for 'head' represents both the concepts of 'leadership position' as well as 'the upper part of the human body', without being disambiguated. In this network, only colexification patterns that appear in at least 3 languages and 3 language families are considered in order to reduce noise due to errors.

- **OmegaWiki**
  The database OmegaWiki (www.omegawiki.org) is an open-source multilingual dictionary based on a relational database. It is a collaborative project of the Wikipedia community to produce a free, multilingual resource in every language, containing lexicological, terminological and thesaurus information. As the project is based on a wiki system, it allows any visitor of the website to contribute. As a consequence, the OmegaWiki database covers many of the world's languages and a vast amount of domain-specific descriptions. The full OmegaWiki database contains roughly 564,000 expressions for 51,000 concepts in over 1,000 different languages. The platform's users contribute to the project by translating English definitions (in the form of descriptions) to another language rather than translating single words. For example, users do not translate the single word 'age' but one of its definitions as in 'A period of history having some distinctive feature' or 'To begin to look older; to get older'. Using this procedure to acquire new data, the error rate of translation due to ambiguous words is low. This way, it introduces an implicit distinction between senses (i.e. meanings) of the same word. Contrary to Clics^3, the OmegaWiki raw data is not organised in a network structure but it includes a set of dictionaries. From this database, a network of identical translations is created following the procedure described above.

- **FreeDict**
  The website FreeDict (freedict.org) collects several open source, free bilingual dictionaries. The FreeDict database contains a collection of 140 dictionaries in over 45 languages. The databases are available in TEI, a XML language to encode human language. Since its origin in 2000, FreeDict has much evolved and now offers both imported and hand-crafted dictionaries in various sizes. Similar to OmegaWiki, FreeDict is an open-source project which allows everyone to contribute. On one hand, this allows the database to grow quickly and to cover a greater number of words. The quality of the translations, however, is not checked by experts, which might decrease the overall quality. The database of identical translations based on the FreeDict database is constructed in the same way as with the OmegaWiki database, and subsequently deployed to build the third colexification network [DNPG21].

Table 2.1 compares the basic statistics of the three colexification networks that we consider. It is noteworthy that the Clics^3 network contains the least number of words (i.e. nodes), but features the most languages. Contrary to OmegaWiki and Freedict, Clics^3 is more interconnected as it contains more links per node. This can be attributed to the fact that Clics^3 has been constructed from sources crafted by experts. On the contrary, the OmegaWiki and Freedict networks, which were constructed computationally, are much larger. As a result of their size, however, they are less sophisticated: due to the limited selection of bilingual dictionaries, OmegaWiki and Freedict only cover a fraction of the number of languages Clics^3 is covering. Moreover, the construction techniques influence the expected number of errors in the database: we expect OmegaWiki and FreeDict to contain more noise and errors than Clics^3 due to faulty edges.

	Clics^3	OmegaWiki	FreeDict
nodes	1,647	10,323	27,939
links	4,228	13,691	70,839
languages	2,271	166	19

Table 2.1: Basic statistics of the colexification and identical translation datasets. Clics^3 features the lowest number of nodes, which in this case represent concepts, but it has the highest coverage of languages. On the opposite, FreeDict features nearly 28,000 nodes across few languages. OmegaWiki constitutes a middle ground between the two previous datasets. Nodes in the latter two cases represent words and not concepts, since these are identical translation databases.

As anticipated, colexification and identical translation patterns are collected across languages. In particular, the more languages they feature, the more universal properties of language they can highlight. Nodes in those networks are language-independent. That is, nodes in Clics^3 represent concepts, which are independent from language. Indeed, the concept for 'the gas that surrounds the earth and forms its atmosphere' is named 'air' in English, 'Luft' in German, 'aria' in Italian and so on. Furthermore, nodes in OmegaWiki and FreeDict represent words and their translation into many different languages. Thus, the same node could be identified by different translations of the same word, as for example by the words 'air', 'Luft' or 'aria'. However, in order to deploy these databases for text analysis, a reference language for the nodes is needed. That is, the concepts and words the nodes represent need to be expressed in the same language. Such language has to match the language in which the analysed texts are written. Thus, in this work we use English as reference language, since we will develop methods for the analysis of English texts.

2.1.3 Applications of colexification

The study of colexification and its applications has started in recent years. In particular, the first studies dealt with problems related to linguistic analyses, in particular in the field of historical linguistics. More recently, colexifications have started being of interest of a more variegated set of researches, who started studying the possible applications of the idea of colexification in interdisciplinary fields. Even before the concept of colexification was deployed for interdisciplinary analyses, networks of polysemies were proposed to study the universality of human conceptual structure [YSS$^+$15]. This study proposes a new method to construct semantic networks of polysemous words based on dictionary translations. Clusters of concepts in these word networks suggest that some concepts are more prone to polysemy than others. Similar patterns appear across different language groups. Analyzing the resulting semantic networks relative to different languages, the study's results confirm that conceptual structures referring to nature and landscape are universal in human language. Moreover, semantic networks have a common structure across geographic and cultural features. This suggests that neither cultural nor geographical differences have a great impact on the human conceptual representation and on the way we make sense of the world.

Using a similar approach but relying on colexifications, Jackson et al. prove that the emotional connotations associated with different concepts vary across different cultures but maintain a basis of shared meaning [JWH^{+}19]. Indeed, analysing the community structures related to different language families of the colexification network Clics[3], the study finds variations in the meaning of emotion words across different cultures. In addition, they also show that the conceptual understanding of emotions is based on an underlying universal structure. Moreover, all language families examined in the study differentiate emotions primarily on the basis of hedonic valence and physiological activation (arousal). The variation in emotion semantics can partially be explained by geographical aspects, i.e. languages with greater proximity to each other tend to colexify the same concepts while emotional associations in more distant language groups have different colexification patterns for emotional words, i.e. they show variation of their conceptual understanding of emotions. This paper shows that the analysis of colexification networks can lead to meaningful insights in fields different from linguistics, as in this case the research question deals with psychological aspects of the understanding of emotions.

The assumption that colexification is associated with meaning similarity underlies most analyses performed deploying such concept. While this claim has been stated alongside with the original definition of colexification [Fra08]), it has not been experimentally confirmed at scale. The distinction between polysemy and homonomy (see 2.1) poses one threat to the validity of this assumption: while polysemy describes two related concepts using the same word form, homonomy describes two unrelated concepts expressed using the same word form. Since both of these concepts are included in the principle of colexification, we cannot conclude a priori that colexification occurrences hint to semantic similarity. In fact, while polysemous words indicate by definition a similarity in meaning, homographs do not. Therefore, homographs can be considered as a potential source of erroneous word relations when using colexification to infer the semantic similarity between words. In conclusion, it stands to reason that the validity of assuming semantic relationships between colexified concepts is still in need of experimental confirmation.

The first step in this direction has been presented in [DNPG21]. In this paper, the authors analyse whether and to which degree colexification and identical translation occurrences track affective meaning. The outcome of the study highlights that words in colexification networks are clustered on the basis of the three affective dimensions of valence, arousal and dominance. However, the assumption that colexification networks encode word similarity is far from being tested. In fact, affective meaning is only one type of meaning and a more extensive test is needed. Furthermore, in this paper the authors show that identical translations are a valid alternative to colexification, because they encode better affective meaning. This work sets also the basis for the application of colexification networks to text analysis: indeed, the authors show that such networks can be deployed for the expansion of lexica of words, on which many text analysis approaches rely. In detail, considering an affective lexicon and employing the colexification networks, the author show that it is possible to predict the affective ratings of words that are not present in the original lexicon.

2.2 Text similarity

This work aims at creating a method for text analysis based on colexification networks. In particular, we want to define a text similarity measure and apply it to different problems of NLP. In order to do so, it is essential to define a word similarity metric. In the following part, we define the concept of word similarity and analyse its relationship with a similar concept, namely word relatedness. In general, both concepts measure the degree to which words are associated with each other. The difference between similarity and relatedness, however, lies in the scope of each concept and the type of semantic and functional relationships which are considered in determining the association between words. Subsequently, we review the most relevant algorithm to compute text similarity given a word similarity metric.

2.2.1 Similarity and relatedness

One of the most complex problems in NLP is the analysis of semantic similarity between words and texts. A text is a sequence of words, each of which carries information. Words and combinations of words convey a specific meaning in texts [LMB$^+$06]. For humans, the interpretation of the intended meaning of words from their context is usually trivial. Indeed, humans have a common understanding of the meaning of words in certain contexts. This insight, namely that similarity can be treated as a property characterized by human perception and intuition ([Res99], [OBCM08]), is known since the 1960s and has been analyzed repeatedly in later studies [MC91]. For computational models, however, understanding the intended meaning of ambiguous words is difficult [CCSB13]. Therefore, sorting out lexical ambiguity with computational models constitutes a complex task. For example, it seems obvious for humans to interpret the meaning of the word ‘state’ in the context of geography as a nation state. On the opposite, in the context of physics, its intended meaning is different. While this distinction is trivial for humans, it poses a significant problem for computational models processing human language.

One way to approach this problem is to model the semantic similarity between words. Semantic similarity can be defined as the closeness in meaning of different words and concepts. It is important to note that semantic similarity represents a special case of semantic relatedness [Res99]. In fact, while semantically similar concepts are deemed to be related on the basis of their likeness [GF13], semantic relatedness is a more general concept, covering more types of relationships between concepts. Semantic relatedness indicates the 'strength of connection' between two concepts related in a taxonomy by using all relations between them (i.e. hyponymic, hypernymic, meronymic and any kind of functional relations including has-part, is-made-of, is-an-attribute-of, etc.). In other words, semantic similarity defines similar concepts on the basis of how much they are like each other, i.e. is-a relationships. For example, the words 'gradient' and 'slope' can be considered as being semantically similar as well as related. On the contrary, semantic relatedness includes any type of relationship and association between concepts, such as meronymy (e.g. 'mouth' and 'tongue') or antonymy (e.g. 'dark' and 'bright') [BH01]. These relations are not included in the definition of semantic similarity, therefore the pairs 'mouth' and 'tongue', as well as 'dark' and 'bright' do not constitute examples of semantic similarity. The concepts of relatedness and similarity can be extended to the text-level, thus obtaining the ideas of text similarity and text relatedness.

2.2.2 Applications

The computational analysis of the similarity between texts plays an important role in NLP and artificial intelligence (AI) [GF13]. The earliest models and applications, dating back to the 1970s, were applied to finding the most related documents to a given query. This task is known as 'relevance feedback'. One of the solution proposed, known as Rocchio's method [Roc71], represents documents as points in a high dimensional term space and computes the center of a set of documents in such space using centroids. Decades later, text similarity measures found application in the automatic sense disambiguation of words, which was approached using machine readable dictionaries [Les86]. The task has become famous under the name 'How to Tell a Pine Cone from an Ice Cream Cone', which summarise the essence of word disambiguation tasks. Other applications of text similarity methods proposed during this period include automatic word sense discrimination [Sch98] and automatic text structuring [SSMB97].

As the field of NLP in recent years moved to a firm mathematical foundation and became popular [ER04], text analysis measures found application in numerous domains. For example, [ASCPCVS$^+$04] explored the use of text similarity in image retrieval based on Bayesian belief networks. Other applications related to Information Retrieval (IR) explore the usefulness of text similarity in web retrieval (e.g. 'named page finding tasks') [PRJ05] and automatic text categorization [LG05]. In this last application, the method computes the text similarity considering not only the mean but also the standard deviation of the classic vector-based model (as in [Roc71]). Furthermore, text similarity measures are applied to the evaluation of text coherence [LB05] and a variety of similarity measures employing different representations of lexical meaning have been proposed: word-based, distributional, and taxonomy-based methods. Finally, text similarity metrics are also used in machine translation ([LZ04], [KPS04]) and text summarization ([OMMI03], [ER04]). In the latter case, a stochastic graph-based method is used to compute the relative importance of textual units.

Given the ever broader range of applications text similarity models are used for nowadays, we think that the interpretability of the model is a very important topic. Contrary to black-box models, which often rely only on statistical properties of the training datasets, interpretable models allow the developers to identify exactly which signals and properties of the analysed data are used to produce the results. This is especially important in use cases where text similarity models directly impact a decision making process, which might influence people's lives. A fair decision making process can only be possible if every step is transparent, thus making the decision makers accountable for their actions. On the contrary, it has been shown repeatedly, for example in [BGMMS21], that black-box models trained using large corpora of texts tend to be biased towards the hegemonic worldviews inherently present in the training data. This is a model property which should, in our opinion, be avoided as much as possible.

2.2.3 Drawbacks

While text similarity methods find application in a variety of tasks, they are affected by some drawbacks. In particular, most of the methods developed in the early days of NLP base their computations on the number of different words (i.e. tokens) that occur in both input texts [BLL98]. In such algorithms, texts are represented as term-vectors, i.e. vectors whose entries correspond to the set of unique tokens in the texts. Even though these methods have been improved using methods such as stemming, stop-word removal, part-of-speech tagging, longest subsequence matching, as well as various weighting and normalization factors [SSMB97], they still present fundamental weaknesses. Indeed, different but semantically related terms are not matched and cannot influence the final similarity score [YTPM11]. For example, the similarity between the two term vectors v1 = (read, old, newspaper) and v2 = (study, ancient, pamphlet) is equal to 0 according to such similarity measures, even though pairs of these terms present shared meaning. This is due to the fact that the two vectors v1 and v2 do not have any word in common, even if their words are close in meaning. Most traditional, vector-based similarity metrics will fail to detect any similarity between these vectors.

As a consequence of the reliance of traditional methods on the shared words between texts, these algorithms present difficulties in handling very short text segments or single sentences. In fact, in the case of short text inputs, they are represented by very sparse term-vectors, which results into computational inefficiency as well as unacceptable performance in similarity computation [BLL98]. Indeed, while long texts usually contain at least a few co-occurring words and thus enable the computation of a valid text similarity value, it can easily happen that two short texts do not contain any common word. However, people can express similar meanings using significantly different sentences in terms of structure and words chosen, which is enabled by the inherent flexibility of natural language [LMB$^+$06]. The lack in effectiveness of these methods in processing short texts is critical for the current NLP challenges. In fact, in recent years, social media have seen an increase of interest from the public and the media and subsequently have become one of the preferred ground for scientists to test hypothesis and conduct experiments. On such platforms, interactions are limited to a finite set of possibilities. Among those, text production and likes and dislikes are the most used. The nature of the medium and in some cases rules internal to the social network stimulate the production of very short texts. For example, Twitter imposes a very strict character limitation to texts posted in the platform. As a consequence, the need of methods that perform well on short texts is crucial for the analysis of these heavily studied platforms [BCCNEA20].

Furthermore, traditional methods are language-dependent. Indeed, term-vector representations result to be inapplicable when measuring similarity between documents in different languages, because they present completely different sets of words [YTPM11]. This problem is usually tackled by mapping terms of different languages onto a common concept space, a practice that presents its own flaws. In addition, corpus-based text similarity measures designed for analyzing long texts are very often domain specific. Indeed, they are based on corpora of texts and are very effective when analysing documents that deal with topics belonging to the same domain of the corpora. However, it is not easy to adapt these methods to analyse texts from other domains [II08], since this would require the modification of the underlying concept space. These lacks of adaptability to the language and the domain of the text is related to the complexity of human language and to the consequent inability of computational methods to automatically infer the meaning of a word from its context.

Methods that measure the similarity between texts can be roughly classified into 5 categories: vector-based, knowledge-based, corpus-based, hybrid and descriptive feature-based text similarity. In the following sub-chapters some of the most prominent methods belonging to these categories are introduced, compared and discussed.

2.2.4 Vector-based text similarity

The first text similarity models were based on the assumption that similar texts share a higher number of words [LMB+06]. Technically, this assumption translated into methods based on lexical matching and word co-occurrence [CCSB13]. Often known as 'Bag of Words' (BoW) methods, because they do not rely on the position of words in the text, such methods found application mainly in the field of Information Retrieval [MBK00]. More precisely, these approaches were first used to find the most related text documents for a given input query, known as 'relevance feedback' task [SL68]. Usually, vector-based models deploy a pre-compiled list of words. Aiming at including the majority of meaningful words in natural language, this list can be of great length. These models represent each document as a vector in the high-dimensional space generated by said word list [LMB+06]. Each document is thus reduced to a vector of numeric values where each value represents the word count or the frequency of the word corresponding to the entry [BNJ03]. In the case of relevance feedback tasks, the query is represented with a vector in the same space. In order to determine the similarity between two document vectors or between a query vector and a document vector a vectorial distance function is applied.

In vector-based similarity measures, the semantic aspects of the input documents are represented as a vector. When addressing information retrieval tasks, each position in the vector typically represents a tokenized word or expression [Kow97]. Early vector-based similarity measures differ mainly in the method they use for weighting the vector entries relative to each document. In a binary approach, the presence of a token is registered. In order to determine if the count of occurrences of a token is high enough to be representative for the semantics of a document, a lower threshold is introduced [Kow97]. The vector entries corresponding to tokens which appear in the text more times that the preset threshold are set to 1, while the entries of tokens not deemed as significant enough are set to 0. On the opposite, when using a weighted approach, the number of occurrences of each word constitutes the basis of the weights of the token vectors. One popular approach is the tf-idf scheme [SM86], in which weights are determined by multiplying the vector of the term frequency counts (tf) by the vector of inverse document frequency counts (idf). In this case, the idf is a measure for the specificity of a token in the whole corpus, which accounts for the fact that a match for a rare word in two documents is more meaningful than a match for a common word. Usually, the idf is transformed into a logarithmic scale and suitably normalized [BNJ03]. Subsequently, a distance metric is used to determine the final similarity between two weighted document vectors. Usually, the cosine, Jacard or Hamming distances are used [BH01].

Vector-based models present significant drawbacks. First, as already anticipated, they do not capture the similarity between semantically related terms. Different word forms are simply not matched, regardless of the relationship between them. The flexibility of human language allows to convey very similar meaning deploying different words [LMB+06], for instance with the use of synonyms. However, this aspect does not only affect synonyms, but also grammatical and inflectional forms. For example, different verbal tenses of the same verb are considered as different words, as in 'go', 'goes' and 'went'. In this case, vector-based models are ineffective. Another drawback is related to sentence representation. Since the dimension of the vector space is usually very large, the term-vectors can be very sparse. As a consequence, the models are usually very inefficient in their computations [II08]. In cross-lingual settings, this problem is even more severe because vocabularies of different languages typically present very little overlap [YTPM11]. Moreover, the default removal of function words such as “and”, “for” or “the” can be seen as another drawback, since the presence of stop-words in a sentence can contain important information about its structure. This problem is especially severe when analysing short texts and sentences, whose length is shortened by the removal of such words [LMB+06]. Extensions of traditional vector-based models address some of these drawbacks. For instance, pattern matching methods incorporate structural information about texts and sentences in addition to the traditional word co-occurrence information [JM09]. Such methods are commonly used in text mining [CY05] and in natural language processing systems that carry conversations with users, known as conversational agents.

2.2.5 Knowledge-based word similarity

Knowledge-based approaches to text analysis use semantic networks as foundation to determine the semantic relatedness of groups of texts. Semantic networks are graph structures that represent knowledge with the use of interconnected nodes and arcs [Sow91]. They consist of a set of nodes representing objects (terms or concepts) and a set of directed or undirected edges (known also as links or arcs) representing relationships between the objects. In order to determine the similarity between texts, semantic networks are used to compute pairwise similarity values for the words in the text. Subsequently, algorithms are used to transform these pairwise similarity scores into a total text similarity value. The methods by which this is achieved depends on the task tackled by the model. The usage of a knowledge network as basis for the computation is an approach to model common human knowledge about the meaning of words in natural language. This knowledge is usually stable across a wide range of application areas [LMB+06].

Semantic networks

Knowledge-based models rely on databases that store information about the relations between words or concepts. WordNet [MBF+90] is among the most popular semantic networks used in NLP. It is described as "an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory" [MBF+90]. Originally published in 1990 and developed by the Princeton University, this lexical database currently contains 117.000 sets of synonyms, called 'synsets'. English nouns, verbs, and adjectives are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets [MBF+90], as for example synonymy (similarity of meaning), antonymy (opposite meanings), hyponymy (supertype-subtype relationship), meronymy (part-whole relationship) and morphological relations (different derived forms of a word). Word similarity measures based on WordNet assume that words belonging to the same synset are interchangeable in some syntax [LMB+06] and thus related.

With the aim of overcoming the linguistic limitation of WordNet, which features only English words, two multilingual lexical databases have been published: EuroWordNet [Res95] and BRICO [Haa00]. EuroWordNet is described as "a multilingual database with lexical semantic networks" [Res95]. The structure of the network is the same as the one of WordNet [MBF+90]: words with similar meanings are represented as synsets, which are related to each other using various types of semantic and lexical relations. The network includes several European languages - Dutch, Italian, Spanish, German, French, Czech and Estonian - and can be used, among others, for monolingual and cross-lingual information retrieval. The second multilingual semantic network, BRICO, was created with the purpose of combining translation dictionaries with the implicit ontology present in WordNet. This way, conceptual structures in WordNet are mapped onto corresponding conceptual structures in other languages. The initial version of BRICO featured Spanish, Italian, German, and French translation dictionaries and it was later extended to Dutch, Danish, Swedish, Finnish, Portuguese, and Swahili [Haa00].

The measures of word semantic similarity and relatedness based on semantic networks are manifold. In particular, we can distinguish methods that rely on semantic networks as knowledge basis [HSO98], measures based on information content ([Res95], [Lin02],

[JC97]) and approaches that rely on the path length between words in the base network ([LC98], [WP94]). In particular, most methods knowledge-based methods rely on WordNet and take into account all relationships between words stored in this database. For example, the approach proposed by Hirst–St-Onge [HSO98] explores the idea of lexical chains, i.e. cohesive chains in which a word is included if it bears at least one cohesive relationship to a word that is already in the chain. Cohesive relationships include a wide range of relationships, which can be as concrete as identity or as vague as associations of ideas. Therefore, we can say that these methods measure semantic relatedness. On the opposite, methods based on information content rely only on one type of relation between words. In particular, these methods, which assume that the similarity between two concepts can be judged by “the extent to which they share information” [BH01], usually rely on the notion of least common subsumer, i.e. the most specific concept which is an ancestor of both words when considering hyponymy (is-a) relations [BH01]. Moreover, the third category of approaches is constituted by methods which compute the similarity between concepts on the basis of the length of the shortest path between them. In a study comparing five word similarity measures ([JC97], [Lin02], [Res95], [LC98], [HSO98]) belonging to the three categories previously introduced, the measure proposed by Jiang and Conrath [JC97], which belongs to the category of methods based on information content, was shown to perform best overall [BH01]. Moreover, the study shows that the measure based on WordNet [HSO98], which incorporates the greatest variety of information from the network, clearly performed the worst. Furthermore, the authors concluded that "venturing beyond hyponymy into other lexical relations in WordNet in practice hurt more often than it helped" [BH01].

In this thesis, we present a novel approach where colexification networks are used as a knowledge base for computing the semantic relatedness between concepts and words. While the types of relationships between words included in WordNet are clearly defined (e.g. synonymy, hyperonym), in colexification networks these relationships are not explicitly distinguished. However, colexification relationship can cover all the different kinds of relationships included in WordNet. For example, among the pairs of concepts linked in the colexification network Clics^3, it is possible to find synonyms (e.g. ’true’ and ’certain’), antonyms (e.g. ’day (not night)’ and ’night’), hyponyms (e.g. ’animal’ and ’bird’), and meronyms (e.g. ’feather’ and ’wing’). Therefore, we can hypothesize that the WordNet network and colexification networks overlap. Moreover, due to the great variety of cultural backgrounds of the languages included in colexification networks, the assumption that the relationships in colexification networks include even more nuanced types of relations between words seems to hold. However, due to the restricted scope of Clics^3 compared to WordNet, the number of concepts featured in the first is not comparable to the extensivity of WordNet. On the contrary, automatically built colexification networks from dictionary sources might be of use when approaching tasks which require a high number of words. In this thesis, we will explore the possibilities these smaller but meaningful datasets can grant to NLP research.

2.2.6 Corpus-based text similarity

Corpus-based methods compute the degree of similarity between words and texts using information exclusively derived from large corpora of documents [MCS06]. Such methods mainly rely on statistical information drawn from these large collections of texts. One of the main advantages of using distributional measures from large corpora is that the resulting models cover significantly more tokens than any dictionary-based measure [II08]. The underlying assumption of this approach is that the meaning of a word can be inferred from patterns in its usage, for example from its position in relation to other words. That is, words used in similar ways and in similar contexts are assumed to be related. Corpus-based text similarity measures are among the most popular approaches in computational text analysis. In the following sections, we report some of the most relevant corpus-based text similarity methods which have been developed in recent years.

Latent Semantic Analysis (LSA)

Developed in the field of automatic language processing, Latent Semantic Analysis (LSA) [DDF+90] is a purely statistical method for the evaluation of the similarity of words and texts. In this approach, the relations between words are inferred from their co-occurrences and are extracted automatically without the need to specify rule systems or enter dictionaries in advance. LSA is based on the assumption that the patterns of use of words can be used to infer their meaning. In particular, the method relies on the hypothesis that a frequent use of different words in similar contexts is an indication of similarity in content. Therefore, the aggregation of the contexts in which a given word does or does not appear determines the similarity in meaning of words and sets of words [DDF+90]. For example, food words will co-occur with other food words more often than with words related to the automobile world. LSA is based on the statistical properties of a large corpus of natural language text, which are used to generate a representation that captures the similarity of words and text passages [II08]. LSA tackles some of the drawbacks of standard vector-based models, namely the sparseness of matrices and the high dimensionality of the space. In fact, similarity in LSA is computed in a lower dimensional space, in which second-order relations among terms and texts are exploited [MCS06].

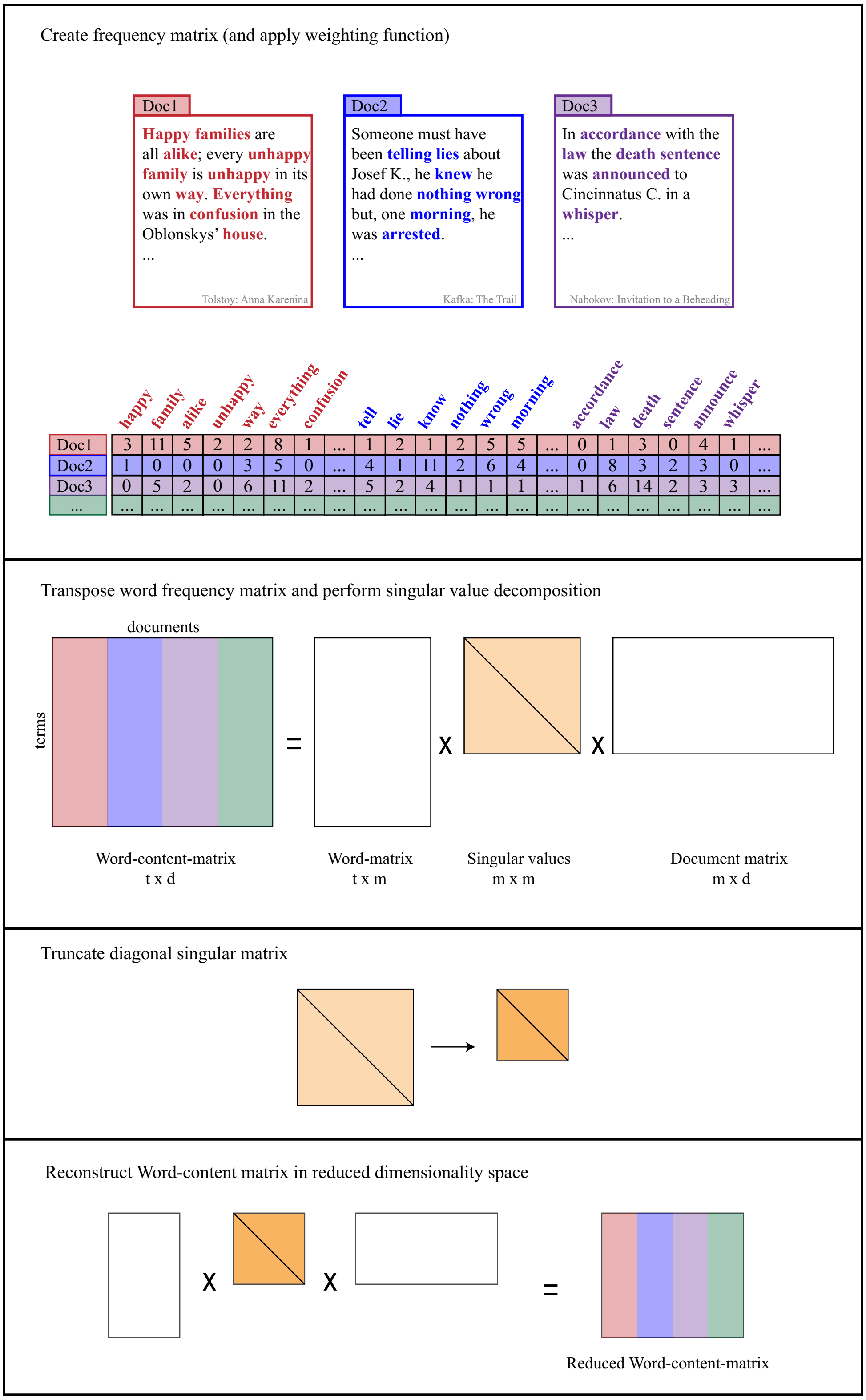
LSA is based on large collections of texts. Usually, the texts are first split into paragraphs of approximately equal length. Then, the relationships between words in the documents are abstractly represented as a frequency matrix, where the columns contain the individual documents and the rows represent the different words. Therefore, the number of rows in the matrix corresponds to the number of unique words in the corpus, which, in the case of large corpora, results in the high dimensionality of the matrix. Each cell of this matrix, called word-context-matrix, contains the frequency of occurrence of a word in a document. In case large corpora of natural language are used, the frequency matrix can be very sparse [LBHS07]. In this case the word-context matrix might become too large to be practically used for further computations. In order to reduce the information to its core content, noisy signal is removed following several steps: first, potentially redundant words such as stop words (prepositions, conjunctions, articles and others) are removed from the matrix. Next, a weighting function is applied to the cell frequencies. This function emphasizes the importance of words occurring frequently in a restricted set of contexts, while it weakens the importance of words which may occur with equal frequency but are evenly distributed. In fact, words that have an even distribution across the texts are considered as not conveying specific information. The third step consists in the decomposition of the frequency matrix using singular value decomposition (SVD), which results in three separate matrices: a word matrix with the factor values of the words, a diagonal matrix of the sorted singular values, and a document matrix. Then, the diagonal singular matrix is truncated by deleting small singular values and thus reducing the dimensionality. Finally, the original word-context matrix is reconstructed from the reduced dimensional space [LMB$^+$06]. This process of dimensionality reduction is illustrated in figure 2.3.

The dimensional reduction ultimately originates a space in which words are distributed according to their co-occurrence with other words. In this space, words are represented as vectors, i.e. they are represented by the part of their content that is manifested through co-occurrences with other words [LBHS07]. When analysing a text, each sentence is represented with a vector in the reduced-dimensional space. In order to transform word vectors into vectors on the sentence or text level, sentences and texts are mapped into the semantic space with a process named 'folding in'. This process consists in the sum of the vectors relative to words that occur in the sentence. Eventually, sentence vectors are summed up resulting in text vectors. The similarity between two texts' vectors can then be calculated using the cosine of the angle between their corresponding row vectors [FKL98], also known as cosine similarity. Let v and w be two vectors in the same space, the cosine similarity between them is computed as:

$$cos(\theta) = \frac{v \cdot w}{||v|| \cdot ||w||}$$

where θ is the angle between the two vectors.

LSA is one of the most powerful and popular methods in computational text analysis. However, it also presents some drawbacks: first, the corpus used for training the model specifies its domain, i.e. LSA is inherently domain specific. In fact, when using LSA on a text not belonging to its training domain, some meaningful words might not be contained in the pretrained vector space. This can lead to problems in the representation of the text in the space and thus in the computation of the similarity between texts [LMB$^+$06]. Furthermore, LSA is not recommended for handling short texts and single sentences. In fact, since its dimensionality is fixed according to the dimensionality of the training corpus, short input segments result in very sparse vectors, which can result in the impossibility to estimate the similarity between them. An additional drawback, which affects particularly theoretic works about semantics and word relationships, lies in the fact that LSA is not linguistically interpretable. In fact, contrary to many other text



Create frequency matrix (and apply weighting function)

Doc1

Happy families are
all **alike**; every **unhappy**
family is **unhappy** in its
own **way**. **Everything**
was in **confusion** in the
Oblonskys' **house**.
...

Tolstoy: Anna Karenina

Doc2

Someone must have
been **telling lies** about
Josef K., he **knew** he
had done **nothing wrong**
but, one **morning**, he
was **arrested**.
...

Kafka: The Trail

Doc3

In **accordance** with the
law the **death sentence**
was **announced** to
Cincinnatus C. in a
whisper.
...

Nabokov: Invitation to a Beheading

	happy	family	alike	unhappy	way	everything	confusion		tell	lie	know	nothing	wrong	morning		accordance	law	death	sentence	announce	whisper	
Doc1	3	11	5	2	2	8	1	...	1	2	1	2	5	5	...	0	1	3	0	4	1	...
Doc2	1	0	0	0	3	5	0	...	4	1	11	2	6	4	...	0	8	3	2	3	0	...
Doc3	0	5	2	0	6	11	2	...	5	2	4	1	1	1	...	1	6	14	2	3	3	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Transpose word frequency matrix and perform singular value decomposition

Truncate diagonal singular matrix

Reconstruct Word-content matrix in reduced dimensionality space

Figure 2.3: Dimensionality reduction using singular value decomposition as implemented in the LSA model.

similarity measures, LSA does not allow for any deep insights into the reason why some terms are deemed to be similar [CM05].

Hyperspace Analogue to Language (HAL)

Hyperspace Analogue to Language (HAL) [BLL98] is a lexical semantic model based on corpus analysis. Its outputs, based on vector representation, can be interpreted on the semantic, grammatical and abstract level [BLL98]. Similar to the previously described LSA model, this model computes a semantic space on the basis of word co-occurrences, statistically inferring semantic information from a corpus of text documents. In the semantic space, texts are represented as vectors and the text similarity is quantified using a distance metric.

The main difference between HAL and LSA consists in the way the semantic space is constructed. Unlike LSA, which builds an information matrix of words belonging to text units or paragraphs, HAL constructs a word-by-word matrix based on word co-occurrences within a moving window of a predefined width [LMB$^+$06]. Thus, a vector in the semantic space representing a word stores the entire history of that word in the context of other words. In particular, the semantic space is constructed as follows: a moving window, often chosen to be of size 10 to preserve locality of reference while minimizing the effects of different syntactic constructions [BLL98], registers the co-occurrences of the words within the moving window and stores them in a matrix. Rows in the matrix represent co-occurrences with preceding words in the moving window, while columns represent co-occurrences with following words. In particular, the matrix does not store the pure number of co-occurrences, but a numeric value that is inversely proportional to the number of words that separate the pair in the window. This device helps accounting for structural information, since closer neighboring words are thought to reflect an higher degree of the semantic meaning of the focus word [GF13]. For instance, two words separated by a 9-word gap have a co-occurrence strength of 1, while the same pair appearing adjacently would have a strength of 10. Afterwards, the row and column vectors relative to each word are concatenated to collect the information for preceding as well as following words. This results in one vector in the semantic space for each word. Subsequently, vectors representing sentences or texts are created concatenating the vectors relative to their words. Finally, the similarity between vectors is computed using a distance metric such as the Euclidean distance.

Drawbacks of HAL already became apparent in the experiments conducted as part of the first validation of the method [BLL98]: HAL is not as promising as LSA, especially when analysing short texts. Furthermore, Yuhua Li et. al. in [LMB$^+$06] claim that the word-by-word matrix HAL is based on has problems in capturing sentence meaning. Moreover, sentence vectors become diluted as a large number of words are added to it.

Pointwise Mutual Information with Information Retrieval (PMI-IR)

A different method for the computation of text similarity is PMI-IR. The algorithm deploys Pointwise Mutual Information (PMI) and Information Retrieval (IR) to measure the similarity of pairs of words [Tur01]. Mutual Information (MI) is a statistical measure of association which compares the joint probability of observing word x and word y together, i.e. $P(x, y)$, with the probabilities of observing the two words independently, i.e. $P(x)$ and $P(y)$ [CGHH91]. This comparison quantifies the degree of dependence between the two variables. In fact, if x and y are independent, $P(x, y) = P(x)P(y)$. The probabilities are computed using an IR algorithm on a text corpus. While the simple Mutual Information algorithm (MI) refers to the average of all possible events, the Pointwise Mutual Information approach takes into account individual events. Similar to LSA and HAL, PMI-IR is based on the frequency of word co-occurrences collected over large text collections [Tur01]. Moreover, like the other knowledge-based methods, also this approach relies on the idea that the meaning of a word is characterized by the words which usually appear in its neighborhood.

PMI-IR was originally introduced in 2001 by Turney [Tur01] as a method to recognize synonyms. In that publication, the author evaluates four different versions of PMI-IR using four different IR approaches. The simplest case, called score 1, considers two words as co-occurring if they appear in the same document. A slightly more sophisticated version of this approach, score 2, deems two words as co-occurring only if they appear 'near' each other. Two words are defined to be near if they co-occur in a window of 10 tokens, independently from the order of appearance. While these first two models do not distinguish synonyms from antonyms, the next model, score 3, reduces the similarity score for antonyms. Finally, score 4 takes into account the context in which two words appear in. The authors show that the latter approach yields to the best results in the problem of recognizing synonyms and that it outperforms LSA on the same task.

While PMI-IR was not originally created with the purpose of text analysis, it has been adapted to the study of texts by other projects [WMSS12]. In this case, instead of computing the degree of independence of pairs of words, the independence or dependence of a word relative to a sentence is computed. In [WMSS12] this approach is used for performing sentiment analysis of texts.

Semantic text similarity (STS)

The previously discussed approach, PMI-IR, can be used as a text similarity method on its own or as part of other methods. In particular, the semantic text similarity (STS) method deploys one version of PMI-IR to compute test similarity. STS was proposed in 2008 by Islam and Inkpen [II08] and is based on a combination of semantic and syntactic information. In particular, this model incorporates three parts: a corpus-based measure of semantic similarity, a modified version of a string matching algorithm, the Longest Common Subsequence (LCS) algorithm [AD86], and an optional function for common-word order similarity. The algorithm combines these three similarity measures into one sentence similarity value. The method works as follows: first, it uses LCS to compute the similarity between words within a string. Secondly, the LCS score is normalized according to the text length. Thirdly, the normalized semantic similarity between words is computed using a corpus-based method. Finally, the optional common-word order similarity function is applied.

In particular, the LCS algorithm is included in order to account for string similarity. This algorithm looks for the longest subsequence that two words have in common. In this application, three different LCS methods are taken into account: one that matches the longest common subsequence between two words, one that searches for the longest consecutive subsequence at the beginning of the words and one that looks for the same subsequence within the words. For example, when comparing the words 'subsequence' and 'sufferance', the first is 'suence', the second is 'su' and the third is 'nce'. A final score is computed as weighted mean of the lengths of the three matched subsequences. The second part of the algorithm is constituted by a corpus-based method, Second Order Co-occurrence PMI (SOC-PMI). This algorithm computes the similarity of words and, being based on a corpus, allows for large word coverage. SOC-PMI is built on the previously discussed PMI-IR model and is based on the British National Corpus (BNC) [1] as a source of frequencies and contexts. Contrary to the classical PMI approach, SOC-PMI can compute the similarity between two words that do not co-occur frequently, as long as they co-occur with the same neighboring words. Thirdly, the common-word order similarity function might be applied. This is an optional part of the computation and quantifies the similarity in word order of the common words in the two texts. In particular, the function quantifies how similarly common words are used in both texts. In some cases, word order is not important for the scope of the analysis and this function is omitted.

The core idea behind STS is to find for each word in one sentence the most similar matching in another sentence. In particular, the algorithm follows six steps: first, all special characters, punctuation symbols and stop words are removed and the words are lemmatized. Secondly, all tokens which occur in both texts are counted and removed from the texts. Step three consists in the construction of a string similarity matrix A for each pair of remaining tokens. In step four, a matrix of semantic similarity between each pair of token B is computed. Step five foresees the construction of a joint matrix C consisting of a weighted linear combination of the string similarity matrix A and the semantic similarity matrix B.

$$C = \lambda_1 A + \lambda_2 B$$

Subsequently, a list is created with the maximum row and column value in the joint matrix, i.e. every word from one text will be assigned to the closest word in the second text, in an injective fashion. If one text has more words than the other, the remaining words are discarded. Finally, the values in such list are summed up and added to the count of unique words appearing in both texts, which results in a total similarity score.

[1] https://www.english-corpora.org/bnc/

If the common-word order function is included, a weighted word order similarity value, which has been computed beforehand, is added to the total similarity score. Lastly, the score is multiplied by the reciprocal harmonic mean of the number of tokens in each text in order to obtain a balanced similarity score between 0 and 1.

In the evaluation of STS on a dataset of 30 sentence pairs [LMB$^+$06], the method outperforms the results achieved by other state-of-the-art text similarity approaches. Furthermore, the authors conclude that, when analyzing short sentences, the optional common-word order similarity function should be omitted in order to achieve better results. A potential reason for this evidence is that the word order of a short sentence can vary significantly while not changing its meaning.

Latent Dirichlet allocation (LDA)

One of the most known text analysis methods is the Latent Dirichlet Analysis (LDA), which tackles the problem of finding short descriptions of the members of a collection in order to enable efficient processing of large collections [BNJ03]. Usually, the members of the collection are words and the descriptions correspond to topics present in the text. Therefore, LDA is usually applied to the task of topic analysis, i.e. extracting the topics a text deals with. LDA is a corpus-based bag-of-words approach, thus it does not consider the structure of a text. The corpus chosen for the training of the model specifies its domain. LDA is based on the idea that each document consists of several latent topics, which are formed by a mixture of words. Moreover, words can belong to different sets, i.e. topics and the model computes the probability that each word belongs to a topic. Afterwards, words - and subsequently also the documents such words belong to - are assigned to the topic with the highest possible probability. On the basis of this assignment, the topic composition of a document can be determined. The model relies on the assumption that during the composition of a document, words are sampled according to probability distributions governed by hidden topics [YTPM11].

In general, LDA is based on a repeated random selection of text segments, whereby the statistical accumulation of word groups is recorded within each of these segments. The algorithm thus calculates the topics of the text collection and which words belong to the respective topics. In more detail, LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities [BNJ03]. In the context of text analysis, LDA can be used to classify documents. By running one LDA module for each class, we obtain a generative model for classification. Furthermore, this method finds application in documents modelling and collaborative filtering. [BNJ03]

2.2.7 Hybrid text similarity

Most knowledge-based text analysis methods include information from text corpora as well, i.e. knowledge-based and corpus-based approaches can be combined to form text analysis methods, which are called hybrid models. Here we introduce some of the most important hybrid models.

Measuring the Semantic Similarity of Texts

One such approach is presented by Corley and Mihalcea in [CM05]. In that work, the authors combine a word-to-word metric with a text-to-text metric and show that this method outperforms traditional text similarity metrics based on lexical matching. This approach deploys the semantic network WordNet as a knowledge base for the similarity measure. Furthermore, it uses information from the British National Corpus (BNC) [2] to derive the document frequency counts to incorporate information on the specificity of words. The usage of two different sources of information, a corpus and a semantic network, is proper of hybrid models.

The proposed method functions as follows: the text is tokenized and each token is matched to the corresponding word class set. Different word classes are nouns, verbs, adjectives, adverbs, cardinal numbers. This task is called Parts-of-Speech tagging (PoS). Additionally, an open-class word set is created for each text. Subsequently, words within the same PoS tag but belonging to different texts are matched according to their similarity. For nouns and verbs, the similarity is computed with a measure based on the WordNet network (six different methods were explored). For all the other word classes, the single words are matched with their counterparts and included in the corresponding word class set if a match is found. Words that cannot be matched are discarded. Finally, in order to determine a directional similarity between two input texts, a scoring function is used on the words belonging to the open-class word set, i.e. only on the words which have a matching counterpart in the other text. This function computes the normalized sum of each words' maximum similarity weighted by the inverse document frequency (the weighing accounts for the specificity of a word). Finally, the two directional similarity scores are aggregated into a bidirectional, final text similarity score computing the arithmetic mean.

Sentence Similarity Based on Semantic Nets and Corpus Statistics (STASIS)

STASIS [LMB$^+$06] is considered one of the most established methods for the computation of text similarity. First proposed in 2006, it is a sentence-level measure which aims at solving the problems related to domain specificity and high dimensionality. The STASIS algorithm takes into account semantic relationships between words as well as the word order information implied in the sentences. STASIS is an hybrid sentence analysis method based on a structured lexical database and on corpus statistics. While the inclusion of a knowledge base serves to theoretically model the common human understanding of natural language, the actual usage of words is modelled using an additional, corpus-based approach. Furthermore, the selection of different corpora for the corpus-based approach

[2] https://www.english-corpora.org/bnc/

provides the possibility of adapting the measure to different application areas. STASIS can be used in a variety of applications that encompass text knowledge representation and discovery.

STASIS combines information on the semantic similarity of words as well as word order similarity. Based on WordNet [MBF+90], the semantic similarity between words is computed as a nonlinear combination of the shortest path length between the words and the depth of subsumer in the semantic network. Thus, the model takes into account the relationships between the synsets the words belong to as well as the depth of the word in the hierarchy of synsets in WordNet. In the computation, both these factors are assumed to be independent from each other. The word similarity method used by STASIS was proposed by the same authors in 2003 and achieved a high performance in predicting the similarity between words [LBM03]. Indeed, experiments presented in this paper demonstrate that the measure significantly outperformed all previously published word similarity measures. In order to compute the semantic similarity between sentences, STASIS dynamically forms the vocabulary corresponding to each sentence solely basing on those sentences, instead of relying on pre-compiled word list or pre-computed knowledge. In detail, the first step consists in the creation of a joint word list which contains all unique words appearing in both texts considered. Next, lexical semantic vectors with the same length as the joint word list are created for each text, according to the following rules: if a word appears in a text, the corresponding entry in the relative semantic vector is set to 1. Otherwise, the maximum similarity of that word to all words appearing in the text is computed. If this maximum similarity score is higher than a threshold, the entry is set to this value, otherwise it is set to 0. The usage of such threshold has two main advantages: it reduces noise and allows for the comparison of STASIS to classical lexical matching methods by setting the threshold to 1. This way, a semantic vector v for each text is obtained. This vector has as many entries as the length of the joint word list.

In addition to the semantic similarity, the information content of each term is considered. More precisely, an information measure is constructed on the basis of the relative frequency of each word in the Brown Corpus [3] following the idea that frequent words covey less information than less frequent words. Then, each entry in the lexical semantic vector of each text is multiplied by the associated information content. In more detail, let v_i is one entry in the semantic vector v relative to one text and computed using the semantic similarity between the two words w_j and w_k. Let $I(w_j)$ and $I(w_k)$ be the information content associated to the two words,

$$w_i = v_i I(w_j) I(w_k) \quad \forall i$$

where w is the vector containing the information relative to the semantic relationship between the words in a text and the joint word list. Subsequently, the cosine similarity between vectors is used to determine the semantic similarity between texts. Additionally, STASIS considers information about the order of words. In particular, this information

[3] https://archive.org/details/BrownCorpus

allows to distinguish between sentences that share the same words but carry a different meaning, as in: "the dog chases the burglar" and "the burglar chases the dog". More in detail, the word order measure is computed as the normalized difference of word order, where each text is represented as a word order vector with the same length of the joint word list. In this vector, each entry corresponds to the position in which the word appears in the sentence. Finally, a weighted linear combination of the semantic similarity and word order similarity is computed, and results in the similarity between two texts. When defining the overall sentence similarity, the word order similarity is chosen to be weighted less than semantic similarity. In detail, if S_s is the cosine similarity between the semantic vectors representing the two texts and S_r is the word order similarity, the final similarity is computed as:

$$S = \delta S_s + (1 - \delta) S_r \quad \delta \in (0.5, 1]$$

This way the semantic similarity has a higher weight on the overall similarity because the word order similarity plays a subordinate role in the understanding of texts.

Lightweight Semantic Similartiy (LSS)

Another hybrid method which makes use of semantic networks as well as information from text corpora is Lightweight Sematic Similarity (LSS) [CCSB13]. This method was proposed by researchers at the De Montfort University, in Leicester and tackles the problem of sparse text representation. LSS combines semantic term similarities with a vector similarity method generally used for statistical analysis. Additionally, the method addresses the high computational effort of state-of-the-art methods such as LSA (see section 2.2.6) and STASIS (see section 2.2.7). In the original paper the authors successfully apply the method to the problem of comparing the titles of museum artifacts.

In LSS, WordNet serves as the basis of the knowledge-based part of the method. In particular, the knowledge-based metric computes the similarity between two texts by, first, storing a term vector based on the synsets the words belong to and, subsequently, calculating the cosine similarity between the two vectors. The process starts by cleaning and tokenizing each text segment, removing stop words and identifying the synset corresponding to each word. Subsequently, binary term vectors are created for each text, indicating if a word appears in a text or not. Then, a pairwise similarity matrix is computed for all the terms appearing in either of the two texts. More precisely, the term similarity matrix entries correspond to the maximum path similarity value (based on the shortest connecting path) between the synsets of the compared terms. Then, the similarity values of each term to all other terms, which are stored in the similarity matrix, are added to form a text vector. Such vector describes the similarity of the text to the term similarity matrix. This results in one vector for each text. Finally, the cosine distance between vectors is computed, which results in a final text similarity value.

2.2.8 Descriptive feature-based text similarity

Another class of text similarity metrics is based on descriptive features. In this case, a text or sentence is represented using a set of predefined features [MR86]. In particular, each word in a text is described by its properties in relation to these features. For example, verbs may be characterized by the binary feature ‘active’, allowing values of 0 (for passive verbs) and 1 (for active verbs). A noun may be described by the numeric feature ‘weight’, which rates the weight of the object a noun refers to on a numeric scale. Texts are then characterized by the features of the words which it consists of. In essence, these methods are based on classification and regression algorithms. Very often, it is not possible to explain the mechanisms a prediction is based on, i.e. they behave as black box models. When applying such models to descriptive feature text analysis, the similarity between two texts is computed through a trained classifier. Moreover, some extensions of this method distinguish between different classes of features such as primary and composite features [HKE99]. It has been shown that sophisticated descriptive feature measures can outperform the standard techniques for similarity computation [HKE99]. However, it is very difficult to define an effective set of features the model should base on [II08]. This problem is especially relevant when taking into account abstract concepts and ideas since they are rarely clearly defined in terms of their attributes. For example, it might be difficult to assign a value for the feature ’color’ to words like ’hunger’ or ’sadness’. Moreover, these values might be influenced by cultural bias (e.g. ’sadness’ is represented as blue in some English speaking countries, but this is not universally true). Furthermore, the values of a text in relation to its features can often only be collected manually, which results in tedious and time consuming tasks [LMB+06]. Due to these reasons, this category of text similarity methods results to be impractical and its use is not common.

CHAPTER 3

Methods

In this chapter, we expose the methodological framework used for the construction of a theory-based and transparent text analysis approach from colexification occurrences. In addition, we will provide the methodological basis for the analysis of our method on various linguistic-related hypotheses on different text sources. In particular, we start with the construction of networks from colexification databases, the definition of distance between words and concepts within the network and its theoretical analysis. We then define a text similarity metric that draws from the previous distance. The definition of this measure follows some theoretical ideas, which are explained in this chapter. Finally, we test the word and text similarity measures on a set of evaluating and exploring tasks. The metrics used to quantify the results of the method on these tasks are also reported in this chapter.

3.1 Construction of colexification networks

The text analysis method that we develop is based on the linguistic idea of colexification. Colexification describes cases when two different concepts are expressed with the same word in one language (see section 2.1). We employ also an approximation of this concept, the idea of identical translation. Identical translation happen when two different words in one language are translated into the same word in a second language. Rather than employing databases of the occurrences of these linguistic patterns in their raw form, we construct colexification networks (or approximations thereof in the case of the identical translation databases OmegaWiki and FreeDict). In particular, we transform the lists of edges of each database into a graph $G = (V, E)$, where the vertices V are given by concepts or words and the edges E by the presence of a linguistic pattern linking two vertices. The difference between the colexification database Clics^3 and those based on identical translations (FreeDict and OmegaWiki) is minimal: in the case of Clics^3, the nodes represent concepts and the links denote colexification patterns between pairs of concepts. In the approximated colexification networks based on the FreeDict and OmegaWiki databases, nodes represent words and links correspond to identical translation occurrences between nodes. In both cases, the graphs are undirected since both colexifications and identical translation relations are symmetric.

The origin of colexification occurrences in natural language is manifold: they can arise due to meaning similarity, historical, geographical and cultural phenomena, as well as mere coincidence [DNPG21]. In order to filter out cases in which their occurrences are not related to meaning similarity, we apply the same rule as in previous research [RTG$^+$20], [DNPG21]: in the case of Clics^3, colexifications occurring in less than 3 languages and 3 families are excluded, while for OmegaWiki and Freedict, at least 2 languages are required for the link to be included in the network. This way, we intend to remove noisy information from our database. The reason behind a lower threshold for OmegaWiki and FreeDict lies in the fact that these networks cover less languages than Clics^3.

3.2 Enhancement and merging of networks

Originally built for theoretical linguistic analysis, the colexification network Clics^3 has a form which is problematic for text analysis. Indeed, as anticipated in section 2.1.2, Clics^3 contains numerous concepts which are identified by parenthetical additional information, such as 'blow (of wind)' and 'blow (with mouth)'. This is necessary in order to distinguish between polysemous words, i.e. two concepts which share the same word form but refer to different concepts. While this differentiation is necessary on a conceptual level, the specification of concepts is not suitable for text analysis applications, which are often based on lexical matching. In order to tackle this weakness, we decided to split the nodes corresponding to concepts that presented additional specifications. Furthermore, some concepts in Clics^3 are defined using two or more words, as for example the concepts 'needle tree' and 'tree trunk'. While this would not pose a problem for further text analysis applications, it creates difficulties because it neglects implicit connections between words and concepts. For example, in the original Clics^3 network the obvious link between 'tree' and 'trunk' is missing, while the concepts 'needle tree' and 'tree trunk' are connected. In order to alleviate this problem, we split nodes that are defined by more than one word in multiple concepts and connect all the resulting nodes.

The process of splitting network's nodes is necessary, as said, because the original network does not contain some implicit connections. For example, many nodes contain the term 'tree' (such as 'tree stump', 'needle tree', etc) but a connection in the form of a link with the concept 'tree' itself is missing. Indeed, in the original colexification database, the two nodes 'tree stump' and 'needle tree' are not connected, even though there obviously exists a significant semantic link between them. Moreover, the component of the network in which the concept 'needle tree' appears is not connected to the components which include other tree-related such as 'tree stump' or 'tree trunk'. Figure 3.1 panel (a) illustrates the structure of the Clics^3 network in the neighborhood of the node 'needle tree'. Because of the structure of the network in that neighborhood, the obvious connection between 'needle' and 'tree' is not detected, and, since no connection between the two nodes is found, their similarity is 0. Therefore, we decided to split nodes of concepts which are described by more than one word. In particular, each word forming the label of a concept, excluding stop-words, becomes a node and all the nodes originated by the splitting of the same concept are interlinked. The weight of these links is equal to the sum of the weights of the links entering and exiting from the original node. Moreover, the resulting nodes acquire all the links of the parent concept with relative weights. For example, we split the node 'needle tree' into 'needle' and 'tree', which results in two separate nodes. Then we add an edge between the two newly created nodes 'needle' and 'tree'. The weight of the connection is equal to the sum of the weights of the links entering and exiting from the node 'needle tree', as proxy for the number of languages that have a word for 'needle tree'. This procedure is represented in Figure 3.1 panel (b).

In particular, the splitting algorithm follows these steps:

1. list all concepts from the database that are composed of more than one word;
2. for each concept C composed of more than one word, remove stop-words, digits and parentheses, obtaining the collection of strings S_c;
3. if a string $s \in S_c$ is not already the label of a node in the network, such node is added to the network;
4. all combinations of two items from S_c are linked and the link has a weight corresponding to the sum of the weights of the links entering and exiting from the original node;
5. eliminate from the network the concept C and its links to other concepts.

As a consequence of the node splitting procedure, 413 new relations between concepts are added to the Clics^3 network (which originally had 4,228 links). The number of total nodes increases by 405. The extended OmegaWiki network (originally 13,691 links) has 2,091 added edges, while the FreeDict network (originally 70,839 edges) obtains 1,453 new edges. As you can notice, this procedure involves mostly Clics^3 and OmegaWiki. Indeed, at the end of the procedure Clics^3 acquires 10% more links, while the identical translation databases gain respectively 15% and 2% new edges.

Moreover, we decided to create an additional colexification network by merging the three previous ones (Clics^3, OmegaWiki and FreeDict) into a new one. The colexification network obtained contains the information that was stored in each of the previous single networks. The initial efforts in validating the definition of word similarity using the MEN database (see section 4.1.1) and each of the three colexification networks led to the conclusion that the OmegaWiki and FreeDict networks are clearly less suited as basis of a text analysis tool compared to Clics^3. This can mainly be attributed to the sparseness of the networks automatically built from bilingual dictionaries. In particular, these networks contain few edges relative to their number of concepts. This is a consequence of the small

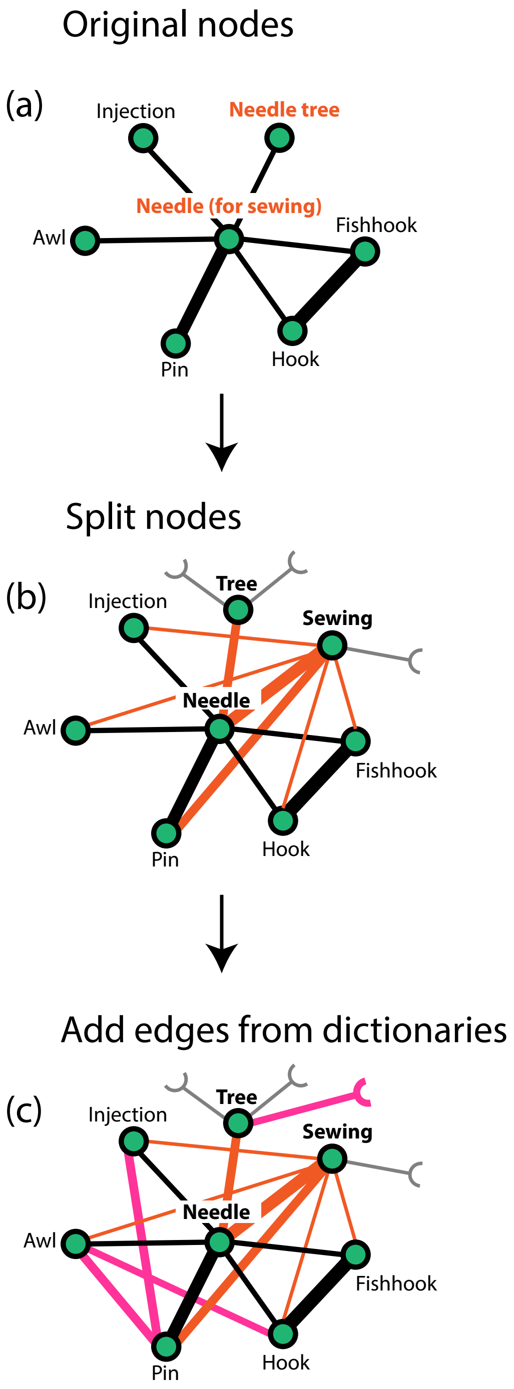


Figure 3.1: (Continues on the following page.)

Figure 3.1: Node splitting process to include implicit connections between concepts in the colexification networks. Nodes that are described by more than one word, e.g. ’Needle tree’ and ’Needle (for sewing)’ (indicated in orange in panel a), are split into new nodes, indicated by each single word in the labels of the parent nodes (stop-words excluded). Panel (a) represent the original structure of the component of Clics^3 including the nodes in the example. Panel (b) represents the structure of the network after the splitting process. New nodes and new edges (orange colored) are created. In particular, ’Needle tree’ is split into ’Needle’ and ’Tree’ and ’Needle (for sewing)’ is split into ’Needle’ and ’Sewing’. The thickness of the links represents the edge weights. After the splitting process, the component of the network is connected to the previously existing nodes ’Tree’ and ’Sewing’, which did not originally belong to that component. Panel (c) represents the addition of edges contributed by the OmegaWiki and FreeDict networks. If two words are contained in Clics^3 and connected in one of the two identical translation databases, the corresponding edge is added to the colexification network (violet edges).

sample size of languages used for their construction. Additionally, due to their size, the implementation of methods deploying OmegaWiki and FreeDict requires significantly greater computational effort, which is out of proportion considering that they achieve significantly worse prediction results in the first experiments. Thus, we decided to use the information contained in the OmegaWiki and FreeDict networks by extending the Clics^3 network through the inclusion of additional edges. In particular, for each pair of concepts already contained in Clics^3, we add all the edges which occur in either the Omegawiki or Freedict networks. This process is illustrated in Figure 3.1 panel (c). This way, a new, combined network based on the Clics^3 set of concepts in constructed. Containing edges from all three colexification networks, this combined network is the most interconnected and thus the most general network. Hereinafter, we only use the combined network.

3.3 Self loops

Colexification patterns deal with the relationship between concepts, however, they do not contain any information about the similarity of a concept to itself. While this issue seems counter-intuitive at a first glance, it is necessary to deal with when building a word similarity metric. Such problem arises due to the fact that in colexification networks we only consider concepts and discard words, i.e. these networks are built by projecting the information on the concept space, accounting only partly for the information on the word space. In particular, colexification databases collect patterns that link the space of words to the space of concepts. However, when constructing a colexification network we project these complex relations on the concept space, discarding the word space. Thus, we lose part of the information collected in the original database. In particular, the existence of words that convey only one concept is not represented in a traditional colexification network. We tackle this loss of information adding self-loops representing the lost connections. Figure 3.2 depicts this process.

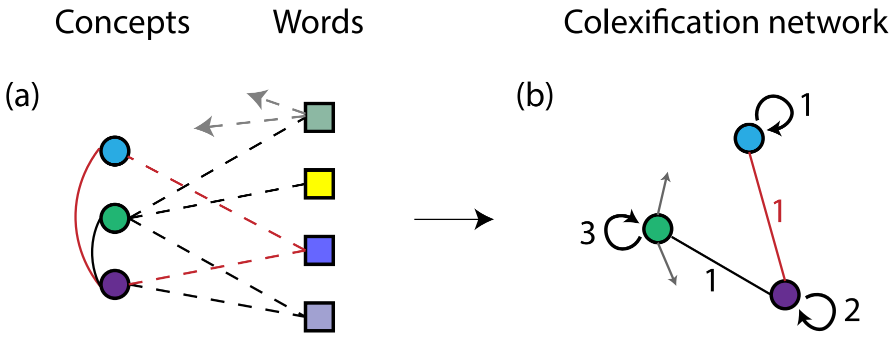


Figure 3.2: Addition of self-loops to the colexification network in order to account for information lost when projecting the colexification data on the space of concepts. In particular, words that address only one concept as the yellow word in panel a, are not taken into account in a traditional colexification network. We add self-loops to the colexification network, as illustrated in panel b, to account for this.

In colexification networks, concepts are linked if a colexification pattern occurs between them. This idea is illustrated in Figure 3.2 panel (a), where the dashed lines connect concepts (illustrated as circles) to the words (represented as squares) they are expressed with. If two concepts are expressed using the same word in one language, i.e. multiple dashed lines are connected to the same word, an instance of colexification occurs. This is illustrated by the curved solid lines. For example, in the Figure one instance of colexification happens between the light blue and violet concepts. In fact, these concepts are both conveyed by the blue word, as indicated by the red dashed lines, therefore there is a colexification pattern between the two concepts. The traditional construction of a colexification network from a colexification database is obtained by projecting the connections between words and concepts on the concept space, as represented in panel (b). This way, only concepts are present in the network as nodes and the links represent colexification patterns. However, doing so we lose the information on the relationship between concepts and words. That is, if a 1-1 relation between word and concept exists, i.e. a concept is conveyed by a word that expresses one and only one concept, when projecting on the concept space the relation is lost. For example, in panel (a) the relationship between the green concept and the yellow would be lost. Therefore, we add self-loops to all the nodes in the colexification network, which represent the projection of relationships between the concepts and the words they are translated in, i.e. the existence of words that convey that concept. In other words, the self-loops are given by all the 2-step paths that start and end in the same concept. We weight these links by the number of words present in the database that convey such concept. The same reasoning can be applied to networks built from identical translation databases. That is, also in this case every node in the network will have a self loop.

To sum up, we modified the colexification network in order to suit a definition of distance on its structure. In particular, we split words in order to make implied connections explicit, we merged the three colexification networks into one, based on the set of nodes of Clics^3 and we added self-loops to account for the similarity of a concept to itself. In the next section, we define a word similarity metric on the colexification network obtained.

3.4 Word similarity metric

In order define a text analysis method based on colexification networks, we need to start by defining a word similarity metric. This metric is based on the distance between nodes in the colexification network and takes into account features such as the link weights and the overall structure of the network. Note that the distance between two nodes is the inverse of their similarity. In other words, when the distance is close to 0 the two words are very similar, i.e. their similarity rating is high. On the contrary, when the distance is high the two words are not similar, i.e. their similarity values will be close to 0. In order to theoretically test the metric, we set some axioms that the similarity metric must satisfy in order to comply with the mathematical properties of distance measures and with theoretical assumptions on word similarity. Subsequently, we test the performance of the metric on a database of words rated according to their similarity. Once all the tests have been successful, we transform the word similarity metric into a text similarity measure.

3.4.1 Mathematical axioms

The word similarity metric is based on the distance between concepts in the colexification network. In particular, the higher the distance, the less similar two concepts are. The mathematical definition of distance states that a distance function $d(x, y)$ must satisfy the following criteria:

- identity of indiscernibles, i.e. $d(x, x) = 0 \quad \forall x$
- non-negativity, i.e. $d(x, y) \geq 0 \quad \forall x, y$
- symmetry, i.e. $d(x, y) = d(y, x) \quad \forall x, y$
- triangular inequality, i.e. $d(x, z) = d(x, y) + d(y, z) \quad \forall x, y, z$

In particular, we want our word similarity metric to satisfy the first two axioms (identity of indiscernibles and non-negativity) but not the criterion of symmetry and the triangular inequality. Indeed, we abandon the symmetry criterion in order to represent the idea that it is generally easier to move from a particular instance (i.e. weakly connected node) to a general concept (i.e. strongly connected node) rather than the opposite. That is, this models the idea that in semantic networks some directions are naturally faster (and shorter) than others. For example, we hypothesize that, in a hierarchical structure like the one representing carnivorous mammals (inspired by [CRM99]), it is easier to reach the word 'Carnivora' from the word 'Otter' than vice versa. Therefore, d('Carnivora','Otter') $\leq$ d('Otter','Carnivora') (see figure 3.3). Since we discard the symmetry criterion, it will not satisfy the triangle inequality either. This is again a consequence of the idea that in the network some privileged routs exist.

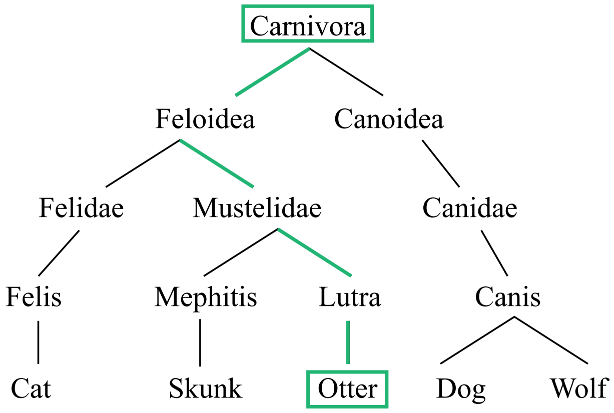


Figure 3.3: Hierarchy of carnivores adapted from [CRM99]. Given this hierarchical structure, it is intuitive to think that it is easier to move from general to particular concepts than vice versa. For example, the distance between 'carnivora' and 'otter' is lower than the distance of the opposite path: from 'otter' to 'carnivora'. This intuition lead us to discard the symmetry and triangular inequality properties of a mathematical distance.

In addition to the mathematical criteria that define a distance function, we set some additional criteria that the distance metric has to satisfy. In particular, these criteria express empirical assumptions that we make on the structure of the network and the properties of word similarity.

3.4.2 Empirical criteria

In addition to the axioms that define a distance function in mathematical terms, we consider some additional criteria that the word similarity function must satisfy. These criteria are empirical, i.e. they originated from theoretical assumptions on word similarity and are inspired by [GvdHAK$^+$14]. In Figure 3.4 the four empirical criteria are illustrated.

These criteria are described as follows.

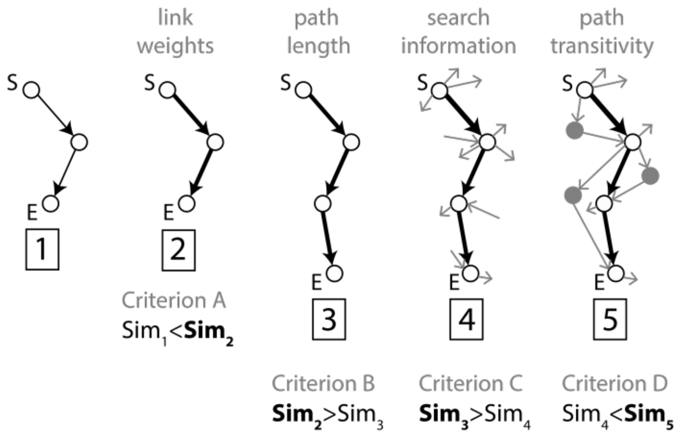


Figure 3.4: Empirical criteria (A-D) for the word similarity function adapted from [GvdHAK+14]. In all these cases, the similarity between node S and node E is computed and the properties of the graphs (edge weights, path length, search information and path transitivity) are taken into account. These criteria have been inspired by the theoretical properties that a word similarity function should satisfy and are inspired by [GvdHAK+14].

Criterion A: the similarity metric must consider edge weights. In particular, strongly connected nodes must be deemed to be more similar than nodes connected by edges with less weight. That is, the distance between node S and node E in graph 1 in Figure 3.4 is lower than the distance between the same nodes in graph 2, where the thickness of the edges represents their weight.

Criterion B: the length of the path connecting two nodes plays an important role. That is, if the weights are constant, shorter paths between two nodes lead to a shorter distance. In Figure 3.4, the distance between the nodes S and E is higher in graph 2 than in graph 3 because the path connecting the nodes is shorter and the link weights are the same.

Criterion C: search information must be considered. That is, outgoing edges along a path weaken the similarity since the information along said path is dispersed. In other words, given a path with same link weights and same length between two nodes, the path with less outgoing edges along the way should lead to higher similarity. In particular, in Figure 3.4, the similarity between node S and E is higher in graph 3 than in graph 4. In fact, in graph 4 the many outgoing edges weaken the similarity between the two nodes.

Criterion D: path transitivity, i.e. the distance should take into account all the ways a node can be reached. Particularity, this criterion states that not only the shortest, but all possible paths between two nodes influence the final node similarity. With an increasing number of possible paths between two nodes the similarity should increase. In Figure 3.4, this case is represented by the fact that the distance between node S and E is higher in graph 5 than in graph 4. Indeed, in graph 5 there are multiple paths that connect the two nodes, therefore there are multiple ways to reach node E from node S.

Once the mathematical and empirical criteria have been formulated and discussed, we define a word similarity metric on the colexification network. We then analyse whether it satisfy all the conditions just described.

3.4.3 Word similarity metric

The similarity metric that we apply to the colexification network is based on [WBS09]. Originally developed to estimate people's trust in others from social network data and inspired by the Google Pagerank algorithm [PBMW99], it reveals to be a suitable word similarity metric on colexification networks.

In particular, we call G a colexification network in which nodes represent concepts and links represent the strength of the semantic connection between them (i.e. language or family weight). Let T denote the adjacency matrix of the colexification network G, where T_{ij} represents the weight of the connection between the nodes i and j, if any. U is the normalized matrix obtained from T:

$$U_{ij} = \frac{T_{ij}}{\sum_{k \epsilon N_i} T_{ik}}$$

Where N_i is the set of neighbors of node i. In order to satisfy criteria A to D (see section 3.4.2), the metric must consider direct links as well as indirect links between nodes. While direct connections are contained in the network's adjacency matrix, indirect ones have to be computed. We compute the indirect strength of the paths between two words i and j based on the direct 1-step path that connects them (if there is any) but also based on the links between the neighbors of i and j. S is the matrix of the combination of indirect and direct connections between nodes, i.e. the similarity matrix and is computed as:

$$S_{ij} = U_{ij} + \beta \sum_{k \epsilon N_i} U_{ik} S_{ik}$$

Where the parameter β functions as a dampening factor. Given $\beta \in [0, 1)$, the impact of nodes far away in the network is weakened. In matrix notation, the equation can be written as:

$$S = U + \beta U \cdot S$$

We can derive:

$$S = (I - \beta U)^{-1} U \tag{3.1}$$

There exists a unique, non-trivial solution to this equation if $\lambda(\beta U) < 1$. Since U is stochastic, i.e. $\lambda(U) = 1$, and $\beta \in [0, 1)$, it follows that $\lambda(\beta U) < 1$.

In case the nodes i and j are not linked, i.e. $T_{ij} = 0$, the similarity of i to j is entirely based on how similar the neighbours of i are to j. On the contrary, if the nodes i and j are neighboring, i.e. $T_{ij} \neq 0$, the similarity of i to j will not only take into account the direct link between the two nodes, but also the similarity between the neighbours of node i and node j. The k-th power of the adjacency matrix of a graph gives the number of walks of length k between any two nodes in the graph. Similarly, the k-th power of the matrix U corresponds to the sum of the products of the weights along all walks of length k in the colexification network. The longer the walk, i.e. the higher k, the stronger the discount (since $\beta < 1$). In other words, long paths have a weaker influence than shorter paths on the final similarity rating.

We can also express $(I - \beta U)^{-1}$ as a geometric sum:

$$S = (I - \beta U)^{-1} U = \sum_{k=0}^{\infty} (\beta U)^k U = U + \beta U^2 + \beta^2 U^3 ... \tag{3.2}$$

Therefore,

$$S_{ij}^{(k+1)} = U_{ij} + \beta \sum_{l \epsilon N_i} U_{il} S_{il}^{(k)} \quad \forall i, j$$

This formula allows for the computation of the similarity between nodes i and j. Furthermore, for a given node i, the computation of the similarity of i to a selected amount of other agents j, if well chosen, will be sufficient, as the similarity to distant nodes in the network is damped out.

3.4.4 Test of the empirical criteria

Once we defined a similarity metric on the colexification network, we test if it satisfies the empirical criteria exposed in section 3.4.2. As explained, these criteria are inspired by theoretical properties of similarity of concepts and by the work [GvdHAK+14]. In order to confirm these assumptions, we manually build five small test-networks with dummy values as edge weights and compute the similarities between pairs of nodes. These networks are a representation of the examples illustrated in Figure 3.4. Given the small size of the networks, it is computationally feasible to compute the inverse of the matrix. That is, instead of approximating the similarity value with the recurring formula 3.2 we analytically compute the similarity matrix with the exact formula 3.1. As a result of this initial analysis, we confirm that for the five test-networks all four criteria are satisfied for β values from 0.3 to 0.9. If β is too small, i.e. too close to 0, the dampening effect is too weak, while in the case of $\beta = 1$ the matrix is not invertible and the similarity values cannot be computed. We choose to set the parameter $\beta = 0.8$ in accordance with insights from the original trust metric [WBS09] and the Pagerank algorithm [PBMW99].

Subsequently, we apply the similarity metric to the colexification network, which results in a similarity matrix suitable for further text analysis tasks. We first validate the metric on a word similarity task in order to calibrate the measure and introduce some modifications to the original trust metric.

3.4.5 Modifications to the word similarity metric

Using the colexification network as basis for the trust metric, we obtain a similarity matrix which quantifies the similarity between concepts on the basis of their distance in the colexification network. We first test the method in a task of word similarity prediction to calibrate the algorithm. In particular, we use a word similarity dataset, the MEN dataset, to perform this first test. The MEN dataset, described more in detail in section 4.1.1, collects word pairs annotated on the basis of their similarity. It consists of a list of pairs of words with a corresponding ground truth similarity value. Contrary to other datasets used in similarity prediction tasks, the MEN dataset rates not only the similarity between words, but also their relatedness. These ground truth values are computed by aggregating ratings given by human annotators.

In this phase, we experiment with different settings of the metric in order to improve the prediction of word similarity. In particular, this analysis serves as the basis for configuring the similarity prediction model. Indeed, being the text similarity metric based on the metric at the word level, the development of a meaningful word similarity metric is important for the whole project. This first experiment shows that the similarity matrix has significant room for improvement, therefore we apply and test several modifications. At the end of this process, we obtain a reliable word similarity predictor, which we will apply to more articulated tasks.

Firstly, we analyse the performance of the method in correspondence to the selection of the parameter β. Such parameter of the trust metric (see section 3.4.3) is the dampening factor for the influence of distant nodes on the computation of similarity. The experiments done on the MEN dataset support the choice of $\beta = 0.8$, which was informed by insights in previous related literature ([WBS09], [PBMW99]) and confirmed by the test of the empirical criteria (see section 3.4.4). In particular, we computed a word similarity prediction task for each parameter beta and concluded that a value of β equal to 0.8 yields the best results. This decision is also in agreement with the finding of the Google page-rank paper [PBMW99], where the authors stated that a β around 0.75 is optimal in most applications. Furthermore, also the authors of [WBS09] identify $\beta = 0.8$ as the optimal value for this parameter. Thus, we will continue using this value of the parameter β in the following applications.

Secondly, we analysed the distribution of the similarity values in this first experiment, finding that the distribution is very skewed. Indeed, most of the similarity values are very close to 0. This distribution is not optimal because the difference in word similarity of different pairs of words is very close and it is hard to establish differences between pairs of words. As a consequence, we decided to transform the similarity values logarithmically to account for this. In particular, the logarithm of the similarity values (which range from 0 to $+\infty$) allows us to better distinguish between values close to 0.

Thirdly, we decided to discard similarity values that are smaller than a threshold. This cleans the similarity matrix of noise, i.e. pairs of words which are very weakly connected and thus do not add any valuable and reliable information to the model. In particular, we set all the values in the similarity matrix which are smaller than the threshold to -1. This way, these words are treated as if they were disconnected and no assessment on the similarity of the pairs of words can be made. We tested different values for the threshold according to various quantiles of the distribution of the similarity ratings: 0.10, 0.25, 0.50, 0.75, 0.90. The results corresponding to a threshold of 0.5 proved to yield the best results on the MEN dataset. Therefore, we set all similarity values lower than the median to -1, which disregards them in further computations using the similarity matrix.

After the calibration of the word similarity metric and the application of the previous modifications to said function, we proceed to define a text similarity algorithm based on that metric.

3.5 Text similarity

Based on the word similarity metric described in the previous section, we construct a metric to compute the similarity between texts. In particular, the challenge of this task consists in finding a method to aggregate a measure that operates on word-level, the word similarity metric built from the trust metric, into a measure at the text-level. In order to do so, we compare two methods: a naive and a more advanced approach. The naive method computes the text similarity by simply averaging the similarity values between pairs of words belonging to different texts, while the more advanced one matches the most similar pairs of words belonging to different texts and incorporates other kinds of information, as the frequency of words in a corpus. We include an analysis of the advantages and disadvantages of the two methods, which motivate our subsequent choice of using one of the two approaches.

3.5.1 Preprocessing

Before applying any text analysis strategy, we need to insitute a preprocessing pipeline that we will apply to every text to analyse. Using several preprocessing steps we convert the raw text data into a suitable format for the text similarity function. This preprocessing consists in three steps: first, all forms of punctuation are removed. Next, all digits are removed. Finally, we convert all letters to lowercase letters in order to make the texts more uniform.

3.5.2 Naive method

Our first approach to transforming the word-based similarity method to a metric at the text-level consists in a method that averages the similarity values of the k most frequent word pairs between two texts. In particular, the selection of k was imposed by the high computational time that the method would have required if we would have applied it to all the combinations of two words belonging to the two texts after preprocessing. Unfortunately, this method has shown to be unsuitable because of some disadvantages. In particular, the method lacks computational scalability, that is the computation of the distance between all pairs of words scales quadratically with the length of the texts, which makes the approach highly inefficient. This lead us to select only k pairs of words, which poses a significant problem. Indeed, selecting the word pairs composed of the most frequent words in each text results in a representation of a text that relies only on its most common words, which has shown to be unsuccessful when testing the method on some text analysis tasks. Selecting word pairs according to their rarity in the chosen corpus would alleviate this problem. However, this solution would privilege text styles that deploy more elaborate language, always scoring them as the most similar texts, independently from the style of the second text. Furthermore, normalizing the rarity score by taking into account the frequency of each word in the text seems to solve this problem, however it greatly diminishes the influence of the text containing fewer rare words in case such a discrepancy exists. Generally, this method based on averaging a set of similarity values corresponding to a list of word pairs proved to be too primitive. Therefore, we decided to continue with a more elaborate method.

3.5.3 Advanced method

Since the naive approach did not yield the wished results, we decided to define a more advanced method. This method for aggregating word similarity scores into a text-level value was inspired by [II08]. In contrast to the previous method, this approach considers not only the output of the word similarity metric but also the corpus-based frequency of words. The frequency is retrieved from the Google Ngram [1] database, which collects the number of occurrences of words in Google Books, a collection of textual sources published between 1500 and 2019. In this approach, the frequency data is used to adjust the similarity values such that rarer words have a higher impact on the resulting text similarity score.

In particular, the computation of the similarity of two texts is represented in Figure 3.5 and unfolds as follows: first, the two pre-processed texts are checked for words which appear in both texts. Those words are then stored in a dataframe *sim* together with an annotated similarity score of 1, indicating an exact match, and removed from the two text vectors. This step is useful when considering texts which contain proper names of characters of places. In fact, these names are not contained in the colexification database, but they convey a high grade of information. Next, the texts are lemmatized using the R-package *textstem* [Rin18] and filtered for words which appear in the similarity matrix, i.e. for the next steps only words that are present in the colexification network are considered, while all the other words are discarded. Subsequently, the similarity for all the possible word pairs composed of one word from each text is retrieved from the

[1] https://books.google.com/ngrams

similarity matrix. These values are stored in a initial matrix with the rows corresponding to words from one text and columns corresponding to words from the second text. Next, an iterative process in applied: in each iteration, the maximum similarity value in the matrix is identified and stored in *sim*. Then, the corresponding row and column is removed from the matrix and the pair of words corresponding to the maximum value are stored. As the highest similarity value is removed from the matrix in each iteration, this method can be seen as a greedy approach to text similarity computation. In simple words, the iteration can be seen as the matching of each word from one text to the closest different word from the second text. This iterative process is repeated until the matrix is either empty or all remaining similarity values are 0. In the subsequent step, information about the frequency and rarity of words according to Google Ngram is incorporated. In particular, for each word pair matched in the previous step, the corresponding frequency value for both words is retrieved from the Google Ngram database. The two frequency values are log-transformed and the average is computed. This results in a weight, named *score*, for each word pair. Finally, the similarity between two texts is computed as the weighted sum of the similarity values of each word pair normalized by the length of both texts. In particular, the similarity between two texts t_1 and t_2 is computed as

$$similarity(t_1, t_2) = \left(\sum_{pairs} sim \cdot score \right) \cdot \frac{m+n}{2mn}$$

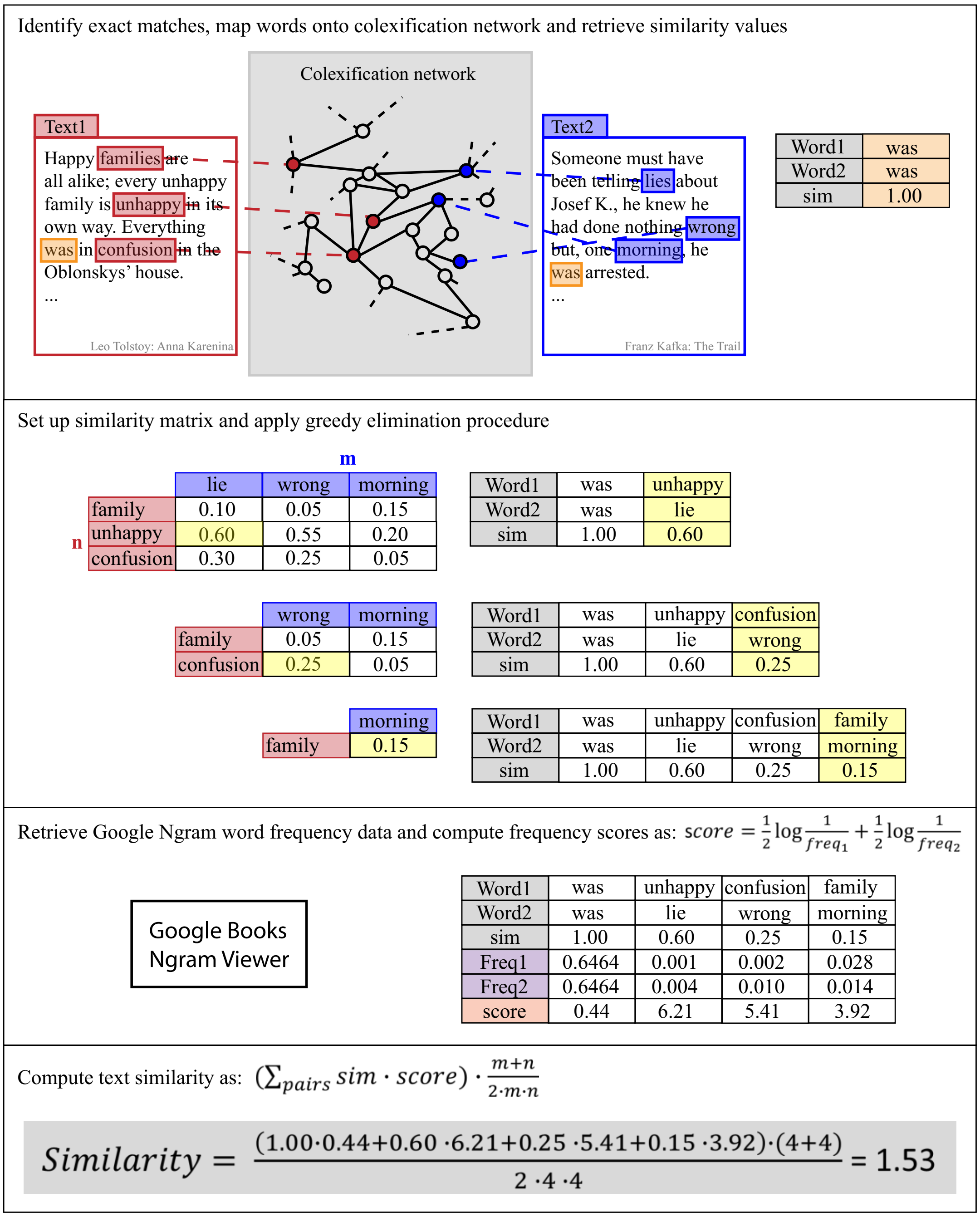
where m is the length of t_1 and n is the length of t_2.

3.5.4 Baseline model

In order to compare the performance of our text analysis tool, we consider a baseline model. The baseline model is based on a tf-idf approach (see section 2.2.4) to embed texts in a vector space. The cosine similarity of the vectors representing the texts determines their similarity. We evaluate the performance of the baseline method using the same approach as for the text similarity model presented in this work.

3.6 Validation and exploration

After the calibration phase, we address with the colexification-based method different text analysis tasks, which we divide into validation tasks and exploration tasks. In particular, validation tasks serve to test the performance of the method in the realm of text analysis and to put the colexification-based approach into comparison with baseline and state-of-the-art techniques. The validation tasks belong to the mainstream text analysis field, and are, for example, word similarity and author recognition tasks. On the contrary, exploration tasks allow us to research the opportunities that our method opens. For example, we will try to extract information on the creativity of people from a word-association exercise. Here, we explain the set up for the experiments and the tools we will use to validate the obtained results.



Identify exact matches, map words onto colexification network and retrieve similarity values

Happy families are all alike; every unhappy family is unhappy in its own way. Everything was in confusion in the Oblonskys' house.
...

Leo Tolstoy: Anna Karenina

Someone must have been telling lies about Josef K., he knew he had done nothing wrong but, one morning, he was arrested.
...

Franz Kafka: The Trail

Word1	was
Word2	was
sim	1.00

Set up similarity matrix and apply greedy elimination procedure

$$\begin{array}{c|c|ccc} & & & m & \\ & & \text{lie} & \text{wrong} & \text{morning} \\ \hline & \text{family} & 0.10 & 0.05 & 0.15 \\ n & \text{unhappy} & 0.60 & 0.55 & 0.20 \\ & \text{confusion} & 0.30 & 0.25 & 0.05 \end{array}$$

Word1	was	unhappy
Word2	was	lie
sim	1.00	0.60

	wrong	morning
family	0.05	0.15
confusion	0.25	0.05

Word1	was	unhappy	confusion
Word2	was	lie	wrong
sim	1.00	0.60	0.25

	morning
family	0.15

Word1	was	unhappy	confusion	family
Word2	was	lie	wrong	morning
sim	1.00	0.60	0.25	0.15

Retrieve Google Ngram word frequency data and compute frequency scores as: $score = \frac{1}{2}\log\frac{1}{freq_1} + \frac{1}{2}\log\frac{1}{freq_2}$

Google Books Ngram Viewer

Word1	was	unhappy	confusion	family
Word2	was	lie	wrong	morning
sim	1.00	0.60	0.25	0.15
Freq1	0.6464	0.001	0.002	0.028
Freq2	0.6464	0.004	0.010	0.014
score	0.44	6.21	5.41	3.92

Compute text similarity as: $(\sum_{pairs} sim \cdot score) \cdot \frac{m+n}{2 \cdot m \cdot n}$

$$Similarity = \frac{(1.00 \cdot 0.44 + 0.60 \cdot 6.21 + 0.25 \cdot 5.41 + 0.15 \cdot 3.92) \cdot (4+4)}{2 \cdot 4 \cdot 4} = 1.53$$

Figure 3.5: (Continues on the following page.)

Figure 3.5: Computation of the similarity between two texts. Text 1 is represented with red color and is constituted by m words, while text 2 is red and has n words. In the first part, words are mapped onto the colexification network and the ones that cannot be matched are discarded. Subsequently, a similarity matrix containing the similarity scores of each combination of word pair is built. The greedy algorithm iteratively selects the highest similarity value in the matrix (highlighted with yellow color) and stores it in the vector sim while deleting the row and column relative to such value. Subsequently, the logarithm of the frequencies of each word is retrieved from Google Ngrams, averaged and stored in a new vector, $score$. Finally, the similarity between the two texts is computed as the mean of the similarity of the word pairs weighted according to $score$ and normalized with the length of the two texts.

3.6.1 Word similarity prediction

In the first validation task, we test the performance of the colexification-based word similarity method in predicting the similarity between word pairs belonging to various word similarity ground truth datasets. After the calibration of the method on the MEN dataset (see section 3.4.5), we test the performance of the method on two new ground truth datasets. The datasets that we take into account are SimLex and SimVerb (see section 4.1.1 for an in-depth presentation). These datasets comprise pairs of words belonging to different parts of speech (for example, SimVerb contains only verbs, while SimLex collects different types of words) together with similarity ratings given by laypeople. In this experiment, we use the calibrated word similarity measure to compute the similarity of the pairs of words included in the datasets. In order to use the same scale for the predicted similarity scores and the ground truth ratings, we normalize the similarity values in the interval [0,1], where a low score indicates a low similarity between the two inputs. Furthermore, the words of the similarity matrix are lemmatized in order to increase the overall coverage between concepts in the network and words in the databases.

The word similarity ratings are predicted as follows: first, all words are lemmatized. Next, we predict the similarity of each word pair by filtering the similarity matrix for the entries corresponding to the words in the pair. Due to the fact that the edges in the colexification network are undirected and the similarity metric is not symmetric, this results in two different similarity scores for each pair of words. In other words, if $s(x, y)$ is the computed similarity between word x and word y, in general $s(x, y) \neq s(y, x)$. Thus, we predict the similarity of the word pair by averaging both values. That is, the final similarity rating sim is computed as

$$sim(x, y) = \frac{s(x, y) + s(y, x)}{2}$$

in order to establish symmetry in the final computation. In some rare cases, a lemmatized word matches more than one row or column of the similarity matrix. For example, the word 'rain' matches with the lemmas of the two words 'rain' and 'raining'. To solve this issue, in such cases the similarity value is computed as the mean of all similarity values retrieved from the similarity matrix. Using this methodology, similarity values for all word pairs that can be matched onto our colexification network can be computed. Word pairs in which at least one word can not be matched to an entry in the similarity matrix can not be predicted and are discarded. In such cases, the algorithm returns a value of -1, which is excluded in the evaluation process. When reporting the results, we consider the coverage of words our method achieves, which is the proportion of word pairs for which a valid similarity score can be computed. The predictive performance of the model is evaluated using the Pearson and Spearman correlation coefficients between the computed and ground truth similarity values. Whereas the Pearson correlation coefficient analyzes the linear relationship between two continuous variables, the Spearman correlation coefficient is based on the rank-ordered values for the individual variables instead of the raw numeric data. The corresponding results can be found in chapter 5. Together with the computed correlation coefficients, we report the relative 95% confidence intervals (c.i.) and P-values. The confidence intervals are obtained by analyzing 200 bootstrap samples of the experiment result data. The P-values in the case of the Spearman correlation coefficients correspond to an approximation of the real value obtained with the R package 'stats'. In the case of Pearson, the reported P-values correspond to the real ones, also computed using the same R package.

3.6.2 Author creativity prediction

As exploration task directly stemming from the word similarity metric, we planned to explore if the word similarity metric is able to predict the creativity of people from their answers to a word-association task. This idea was inspired by a study by Gray et al. [GAC+19] which presents the concept of forward flow on a chain of word associations. Forward flow is a metric that allows to "quantify the conceptual content of naturalistic thought." In practical terms, the forward flow is the average semantic distance between any given thought and all previous thoughts in a chain of thoughts. It can be understood as the ability for a stream of consciousness to flow forward, leaving behind previous thoughts. For example, low forward flow is achieved when thoughts circle back to previous thoughts (e.g., happy, smile, dentist, teeth, smile, happy), while high forward flow occurs when thoughts continue to flow away from the past (e.g., happy, smile, dentist, doctor, hospital, helicopter).

Forward flow uses latent semantic analysis to capture the semantic evolution of thoughts over time (i.e., how much present thoughts diverge from past thoughts). The experiments performed suggest that the forward flow estimate on a chain of word associations predicts the creativity level of the person generating said chain. The study shows that people with high forward flow give creative answers to standard creativity tasks, and those with creative careers (e.g., actors, and entrepreneurs) have higher forward flow than the general population. In addition to creativity, forward flow may also help predict other psychological characteristics, such as emotional experience, leadership ability, adaptability, neural dynamics, group productivity, and cultural success, as well as mental illness.

Our original idea was to use our colexification-based similarity metric to predict the creativity levels of people from the association chains data used in the original experiment [GAC$^+$19]. Unfortunately, it was not possible to access the full data of association chains and the corresponding creativity ground truth scores from the original experiment. Only a small portion of the data was accessible through the repository the team provided. Upon further request, we found out that a large portion of the data was lost. We therefore decided to perform a first analysis with the available data, postponing the work on creativity and reopening it in case the missing data was found. Unfortunately, this has not been the case and we will report only on the study with the partial data.

In particular, we split the word association chains from the original dataframe and considered only subsequent pair of words. We ran the forward flow algorithm on the word pair as well as the colexification-based word similarity method. We consider the creativity score of the author of the word chain and associate it to all the word pairs that appear in said chain. We then compute the Pearson and Spearman correlation between the estimate word similarity and the creativity score. Note that, since low creativity answers to the word association tasks are constituted by pairs of similar words, the estimate of the word similarity will be negatively correlated with the creativity score. That is, low word similarity scores should correspond to high creativity values. We compare the correlation values with the correlation of the forward flow estimate with the creativity score. In this case, the forward flow estimate should be positively correlated with the creativity, since higher creativity corresponds to a higher level of forward flow.

3.6.3 Genre prediction

A validation task for the text analysis algorithm consists in a genre classification exercise based on the Brown University Standard Corpus of Present-Day American English (Brown Corpus) [FK79]. The corpus includes 500 samples of English texts of about the same length (2,000 words) belonging to various genres and published in the same year. We consider the genres in this corpus as classes for the classification task. Since classes in the corpus are unbalanced, we choose to validate our method on the largest sample size possible, i.e. selecting the maximum number of texts from the most frequent genres, thus keeping the classes balanced. In particular, we select the 5 biggest classes in the dataset, which are fiction, belles lettres, learned, lore and press. Furthermore, we perform bootstrap sampling to increase the sample size and show the robustness of the experiments' results. The classification task is evaluated using two slightly different methods, both of which are based on the pairwise text similarity value between each possible pair of texts in the sample set.

Evaluating classification results

We evaluate the performance of the model in the genre classification task using two approaches based on the k-nearest-neighbors (kNN) approach. Here, we distinguish between a classic kNN approach and an advanced kNN approach, which introduces a way of weighing the nearest neighbors of each observation. An illustration explaining how both kNN-based evaluation methods function is shown in figure 3.6.

In the classic kNN approach, the class of a sample is predicted by inspecting the classes of its k nearest neighbors. According to a chosen value for k, the most frequent class that appears in the closest k neighbors to the target text is the predicted class. Since the text similarity metric gives a similarity value for each pair of texts it is easy to define the neighbors of a text and rank them by similarity. Figure 3.6 panel (a) represents this method. This approach takes into account the ranking of texts but not their similarity value. In particular, it does not consider possible gaps in the similarity ratings of texts in the ranking. Therefore, we developed a modification of this approach, which takes into account also this aspect.

Similar to the classic kNN approach, this modification takes into account only the k nearest neighbors of a text. However, this method does also consider the similarity value between a text and its neighbors, not only their classes. The method is represented in Figure 3.6 panel (b). In this case, we compute the mean similarity of texts that belong to each class appearing in the set of nearest neighbors. Subsequently, we assign a score for each class by multiplying the mean similarity with the logarithm of the frequency of this classes' appearance within the nearest neighbors. This way, we consider the frequency of a class as well as the similarity of the neighbors of the text considered. Finally, the class with the highest score corresponds to the class label of the observation. As we can see from Figure 3.6, in some cases the classic kNN and the weighted kNN approaches yield different results.

3.6.4 Classification tasks

Further validation and exploration of the colexification-based text analysis algorithm is done by means of predicting if two texts belong to the same category. This experiments are similar to the genre recognition task, where the definition of category varies across experiments. In particular, we use categories such as authors, books and the year of publication of a book. The datasets deployed for this analysis are the Project Gutenberg corpus (presented in section 4.2.2) and the COHA collection (introduced in section 4.3.1). The experiment is set up as follows: we select some texts from the collection which differ in the variable we want to analyse (e.g. the year of publication), keeping the other variables as fix as possible. Next, we compute all the pairwise similarity of texts belonging to the same class and we compare it with the results of the same amount of computations with texts chosen at random from the other category. Since the text similarity algorithm is based on a random sample of 2000 words from each text, we repeat each computation of text similarity 10 times with 10 different samples. We then evaluate the results using the ROC curve and AUC as explained in section 3.6.4. As the computation of those metrics requires a probabilistic prediction for each ground truth observation in the test set, we frame the experiment as a binary classification task. In this task, we first split the experiment results into a training and a test set. Next, for each element in the test set we predict the probability of two books to belong to the same category (i.e. being

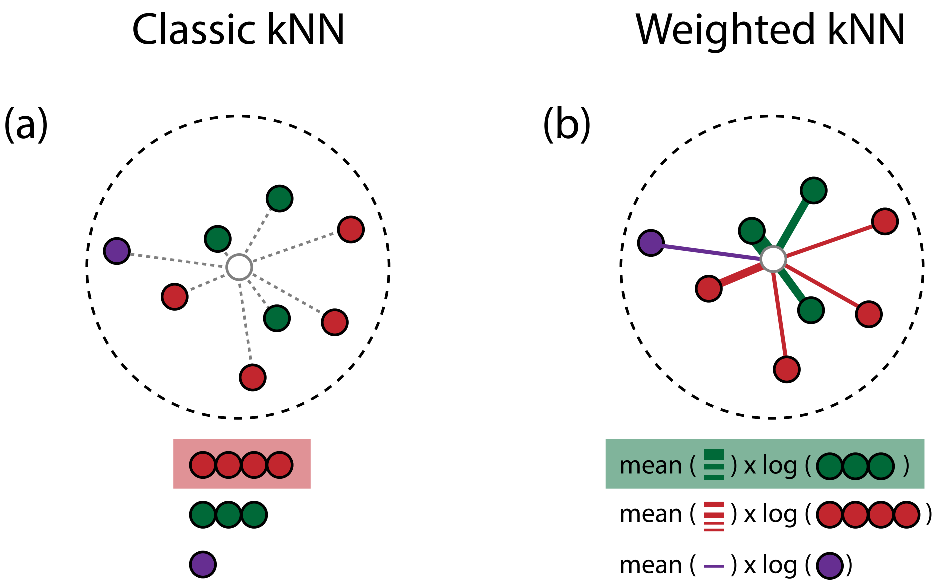


Figure 3.6: kNN-based evaluation methods for the genre-classification task. In this figure, k=8 is considered. Colors represent the genre of each text. Panel (a) shows the classic kNN method predicting the class of an observation as the majority class of the k nearest neighbors of the text considered. In this case, the predicted genre is red since the most neighbors belong to that genre. Panel (b) shows a weighted version thereof: the scoring function is defined by the product of the mean similarity of the neighbors belonging to each class with the logarithm of the number of observations of the corresponding class. Note that the distance of two texts is the inverse of their distance and is indicated in figure by the thickness of the lines. Finally, the class with the highest score is predicted. In the case in the figure, the predicted class is the green one.

written by the same author or belonging to the same book) or to two different ones. This is done using the method illustrated in section 3.6.4. The binary value 1 indicates that two books belong to the same group, the value 0 stands for different groups.

ROC curve

Those experiment are evaluated using Receiver Operating Characteristic (ROC) curves and the corresponding area under the curve (AUC). The ROC curve is a graphic representation of the sensitivity against the 1-specificity of a classification task, given that the predicted values are between 0 and 1. In particular, in our analyses the predicted values correspond to the probability that a text belongs to each of the possible categories. The AUC indicates how well the model can distinguish between positive and negative outcomes, ranging from 0 to 1. The higher the AUC, the better the model can correctly classify the results. However, very high values might hint to overfitting of the model.

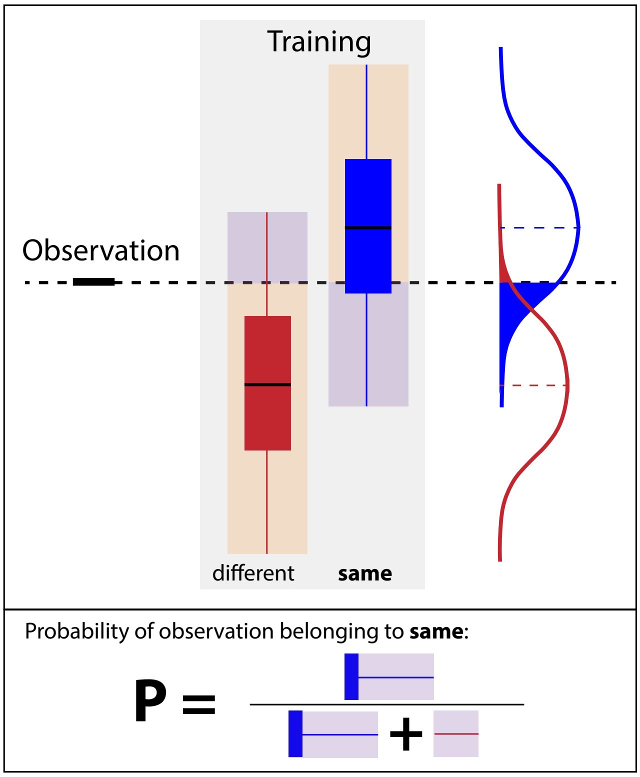


Figure 3.7: Computation of probabilistic predictions for binary classification task. The probability that one observation belongs to the same category of another text is characterized by the proportion of similarities of texts in the training set which belong to the same category (blue boxplot) with a lower similarity than the observed one divided by the sum of this number and the proportion of training observations of which belong to different categories (red boxplot) with have a higher similarity than the observed one. The distributions corresponding to the two boxplots are reported on the right side.

The probability that a text belongs to the same category of a given text is determined by framing this test as a binary classification task. Using 70% of the computed similarity values as a training set and the remaining 30% as a test set, we analyze if the model is able to predict if two texts belong to the same category (value= 1) or to different categories (value= 0). The underlying assumption is that the observations belonging to the same category have a higher similarity score compared to observations belonging to two different categories. The probability that one observation belongs to the same category of another text is equal to the proportion of observations from the training dataset that belong to the same category but have a lower similarity than the one observed divided by the sum of this number and the number of observations in the training set which belong to different categories and have a higher similarity than the observation value. In other words, we compare the distributions of the similarity values of texts belonging to the same category and to different categories as given by the training set and then compute the probability that one observation from the test set belong to one of the two distributions. The highest probability determines the predicted category. This principle is illustrated in Figure 3.7.

CHAPTER 4

Experiments

In the following chapter we describe the datasets used to perform experiments in the course of this project. In particular, we considered tasks belonging to two categories: word similarity prediction and text similarity analysis.

4.1 Word similarity

We use the colexification-based word similarity metric to predict the similarity of pairs of words. The word similarity experiments are based on three of the most commonly used benchmark datasets for this kind of task in computational linguistics. We use one of these datasets, the MEN dataset (see paragraph 4.1.1), to calibrate the word similarity algorithm by iteratively evaluating the model and successively adjusting parameters and implementing new modifications, as explained in section 3.4.5. The other two datasets, SimLex and SimVerb, both addressed in section 4.1.1, were used for the purpose of validation of the calibrated word similarity metric. The main aim of these experiments is to validate the word similarity metric based on colexification networks and thus confirm its suitability as a basis for text analysis applications. In a broader sense, this aims at confirming that colexification networks can be used as a knowledge basis for semantic graphs, yielding competitive results in comparison to other knowledge-based word analysis methods.

4.1.1 Datasets

As mentioned, we use three different datasets in order to calibrate and validate our word similarity prediction model. The MEN dataset 4.1.1 was mainly used for calibration. The reason for choosing the MEN dataset for such a purpose is that it considers semantic similarity as well as semantic relatedness, which is a more general concept. In particular, we repeatedly evaluate the similarity metric on this dataset and adjust various setting and parameters based on the results, e.g. the parameter β or the lower threshold value. For a more extensive analysis of the calibration procedure see section 3.4.5. After completing the calibration process and introducing adjustments to the word pair similarity prediction process, we evaluated its performance on the SimLex and SimVerb dataset with the aim of validating our method. Originally, we considered a third validation dataset, the Stanford Rare Word (RW) Similarity Dataset [LSM13]. However, we found that this dataset is not suitable for our experiments due to the rarity of the words contained. Indeed, only a very small fraction of words featured in said dataset was also included in the colexification network, which led to a very low coverage (smaller than 1%) of word pairs to be predicted. Thus, we concluded that our word similarity prediction model is not suitable for very rare words and decided not to consider this dataset in further analyses.

The MEN Test Collection

The MEN Test Collection (MEN) was originally released in 2012 by Bruni, Tran and Baroni [BTB14]. It consists of 3,000 word pairs obtained through a crowdsourcing platform. In NLP, the MEN dataset is often used to validate algorithms implementing semantic similarity and relatedness measures. One interesting aspect about the dataset is that the developers of the dataset did not distinguish between semantic similarity and semantic relatedness. Indeed, the collection includes "not only pairs of terms that are strictly taxonomically close (cathedral-church: 0.94) but also terms that are connected by broader semantic relations, such as whole-part (flower-petal: 0.92), item and related event (boat-fishing: 0.9), etc." 4.1.1. Thus, the dataset covers different types of similarity and relatedness, which made it very suitable for our calibration purposes. In fact, we want to capture all types of semantic relations between words.

In total, the dataset includes 3,000 word pairs which were "randomly selected from words that occur at least 700 times in the freely available ukWaC and Wackypedia corpora[1] combined and at least 50 times (as tags) in the opensourced subset[2] of the ESP game dataset[3]" [BTB14]. Furthermore, the ground truth ratings were obtained as follows: each auditor was given a list of randomly matched pairs of words. In each case, the annotator had to decide which pair of words was more similar/related. Thus, this dataset is based on comparative judgements of word pair similarity and relatedness, as opposed to absolute similarity score (as in SimLex for example).

Overall, each pair of words was compared and rated 50 times against other pairs of words, which resulted in a score between 0 and 50. The higher the score, the more similar or related is the word pair. Thus, word pairs with a score close to 0 are deemed to be unrelated whereas pairs with a score close to 50 are seen as the most related word pairs in this experiment. In order to control for the quality of the ratings, two individual raters scored each word pairs on a scale from 1 (less similar/related) to 7 (more similar/related).

[1] https://wacky.sslmit.unibo.it/doku.php

[2] https://www.cs.cmu.edu/ biglou/resources/

[3] https://web.archive.org/web/20090106145854/http://espgame.org/

This allowed for the calculation of the inter-annotator agreement, a metric which is often used to evaluate the annotators consistency. The two raters achieved a Spearman correlation score of 0.68, which "suggests that MEN contains meaningful semantic ratings" [BTB14].

The original dataset lists the pair of words and their ratings, as shown in the Table 4.1.

	word1	word2	score
1	sun	sunlight	50.00
2	automobile	car	50.00
3	river	water	49.00

Table 4.1: Format of the MEN dataset. It contains the pair of words and their score, which is between 0 and 50. Here the three pairs with the highest ratings are reported. An high score, close to 50, means that the pair of words is similar or related.

In order to deploy the MEN dataset for the calibration of the model, we normalize the score by dividing it by 50. This way, we obtain a database of word pairs with scores between 0 and 1, whereby lower scores (close to 0) indicate less similar or related word pairs and higher scores (approaching to 1) indicate a higher similarity or relatedness.

SimLex-999

The SimLex-999 dataset (SimLex), first published in 2012, was developed by Hill, Reichart and Korhonen [HRK15]. Compared to other semantic similarity datasets, its main point of distinction is that it explicitly focuses on similarity rather than association or relatedness. Thus, the authors aim to "incentivize the development of models with a different, and arguably wider, range of applications than those which reflect conceptual association" [HRK15]. Another feature which makes the dataset different from other word similarity datasets, is that SimLex includes words belonging to various grammatical categories: concrete and abstract adjectives, nouns and verbs. Each pair is given an independent rating of concreteness and association strength. Due to the diversity of words in the dataset, state-of-the-art methods in the field appear to have difficulties in achieving correlation scores as high as in other datasets [HRK15]. In our validation experiment, we predict the association strength of each pair, which is a measure strongly related to our understanding of relatedness and similarity.

The association strenght scores in the SimLex dataset were gathered as follows: first, an explanation of the difference between similarity and association using various examples was shown to the annotators. Next, a 'checkpoint' question was shown to the annotators to make sure they have the right understanding of these concepts. After that, the annotators were shown groups of 7 word pairs, each of which they rated on a scale from 0 (low similarity) to 6 (high similarity). In total, each participant rated 20 groups of 7 word pairs each. Finally, after averaging, the scores were transformed linearly from the interval [0, 6] to the interval [0, 10]. Analyzing the inter-annotator agreement resulted in a Spearman correlation score of 0.67, which is comparable to similar experiments [BTB14].

The original dataset is illustrated in Table 4.2. It lists not only the ratings and the standard deviation of the association strength of the pairs of words as resulting from the annotation exercise, but also scores according to other databases. In particular, the table contains the POS of the pair of words, their concreteness score as in the University of South Florida (USF) Free Association Norms [NMS11], the quartile of the concreteness scale the pair collocates in, their association ratings according to [NMS11] and a binary number indicating whether the pair is one of the 333 most frequently associated pairs.

	word1	word2	POS	SimLex999	conc(w1)	conc(w2)
1	old	new	A	1.58	2.72	2.81
2	smart	intelligent	A	9.20	1.75	2.46
3	hard	difficult	A	8.77	3.76	2.21

	concQ	Assoc(USF)	SimAssoc333	SD(SimLex)
1	2.00	7.25	1.00	0.41
2	1.00	7.11	1.00	0.67
3	2.00	5.94	1.00	1.19

Table 4.2: Excerpt from the SimLex-999 database. Pairs of words are reported with their association ratings, together with scores from other studies, as for example their concreteness values.

From this dataset, we consider only the word pair and the association ratings, which we normalize in order to make the task comparable to the other word similarity exercises.

SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity

The dataset 'SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity' (SimVerb) was developed by Gerz, Vulić, Hill, Reichart and Korhonen [GVH$^+$16]. Published in 2016, it aims at providing a word similarity evaluation resource which focuses on verbs. The authors consider solely verbs because they "play a critical role in the meaning of sentences" and "these ubiquitous words have received little attention in recent distributional semantics research" [GVH$^+$16]. As its name suggests, the dataset includes 3500 verbs with annotated similarity ratings. Furthermore, the verbs selected to feature in this dataset were chosen on the basis of the USF free-association database [NMS11].

The experiment to gather word similarity ratings was set up as follows: the word pairs were rated using the Prolific Academic crowdsourcing platform[4], an online marketplace

[4] https://www.prolific.co/

very similar to Amazon Mechanical Turk[5] and to CrowdFlower[6]. The experimental setting was inspired by the SimLex dataset (see section 4.1.1): groups of 7 pairs of verbs were shown to the annotators, which were rated on a scale of 0 to 6 by moving a slider. In total, each word pair was rated by 10 annotators. Some pairs were shown more than once to ensure annotators consistency and thus control for the quality of annotations. In total, the experiment included 843 raters which produced over 65,000 ratings. With regard to the inter-annotator agreement, SimVerb achieved a Spearman correlation score of 0.84, which represents "a very good agreement compared to other benchmarks" [GVH+16].

The dataset resulting from this process is illustrated in Table 4.3.

	word1	word2	partofspeech	score	type
1	take	remove	V	6.81	SYNONYMS
2	walk	trail	V	4.81	COHYPONYMS
3	feed	starve	V	1.49	ANTONYMS

Table 4.3: Excerpt of the SimVerb-3500 dataset. Pairs of verbs are rated according to their similarity. The POS and the type of similarity is also included.

In order to proceed with this dataset, we select the pairs of words and normalize the similarity ratings.

4.1.2 Word similarity prediction

We use the SimLex and SimVerb databases to quantify the performance of the colexification-based word similarity method. We do so by deploying the experiment setting explained in section 3.6.1. We then consider the correlation coefficient between the predicted and ground truth similarity values and compare the results with state-of-the-art approaches.

4.1.3 Forward flow

Forward flow can be explained as the ability for a stream of consciousness to flow forward, leaving behind previous thoughts. Inspired by [GAC+19], we aimed at recreating their experiment and thus predict the creativity of test subjects given their word association chains. Unfortunately, not all data necessary was available, thus we performed a modified experiment.

We were able to access the data collecting the word association chains, the forward flow ratings and the creativity values of 3 groups of people: 'Americans', a representative sample of 581 Americans, 'Actors', a group of professional actors, and 'Turkers', a sample of Amazon Mechanical Turk workers. Table 4.4 shows the distribution of data according to the three mentioned groups. As you can see, the sample is imbalanced and the class 'Americans' is the largest, with respectively nearly 6 and 11 times more participants than the 'Actors' and 'Turkers' samples.

[5] https://www.mturk.com/

[6] http://faircrowd.work/de/platform/crowdflower/

	Americans	Actors	Turkers
# Participants	581	52	102
# Word pairs	9,701	930	1,679
Coverage	20%	20%	32%

Table 4.4: Summary statistics of the data for the creativity prediction experiment. The number of participants to the forward flow study for which we obtained the needed data are reported, as well as the amount of pairs of words examined for each class and the coverage, i.e. the proportion of word pairs which could be analysed by the colexification-based word similarity tool.

Each association chain is composed of a vector of 2 to 20 words, a corresponding forward flow value and a creativity rating, which was determined with the administration of various psychological tests to the participants to the original study. Due to inconsistencies in the association chain data, such as missing and repeated words, we decided to treat each pair of consecutive words independently instead of taking into account the full chain of thoughts. The number of available word pairs per group of participants is reported in Table 4.4. The colexification-based approach is not able to compute the similarity for all the word pairs. Indeed, in the case in which one of the words in the pair does not appear in the colexification network, the pair of words has to be discarded because their similarity cannot be computed. The percentage of coverage of the word pairs ranges between 20% to 30%, as reported in Table4.4. Furthermore, we decided not to discriminate between groups of participants but to frame the task as a word pair creativity prediction task. We stored each pair of consecutive words coming from the association chain data in a list and annotated the ground truth creativity value of the corresponding participant. In order to compare the predictive performance of the colexification-based word similarity measure with the forward flow measure presented in [GAC+19], we also collected the individual forward flow distance rating of each pair of words. We did this by retrieving the individual distance ratings from the Forward flow online tool[7]. One extract of the resulting data is shown in 4.5.

	word1	word2	class	creativity_orig	ff_orig
1	toaster	iron	Americans	1.62	0.30
2	iron	mixer	Americans	1.62	0.10
3	mixer	blender	Americans	1.62	0.04

Table 4.5: Excerpt from the final Forward flow data used in the creativity prediction task. Pairs of words are reported with the class of the participant, the creativity value and the forward flow rating.

Finally, we predict the similarity of each word pair and evaluate the correlation of these values with the creativity measures using the Pearson and Spearman correlation

[7] http://www.forwardflow.org/query_data?

coefficients. In our experiment, a high similarity value corresponds to low creativity. Indeed, responding with a word that is very similar to the given seed word in the association chain task corresponds to a low forward flow rating and therefore to a low creativity score. On the contrary, a low similarity corresponds to a high forward flow value and a high creativity score. Therefore, we transform the estimated similarity ratings in order to have them in the same form as the forward flow estimates. The results of this experiment are reported in section 5.1.2.

4.2 Text similarity

We apply the text similarity measure described in section 3.5 to three different datasets in order to validate the measure end explore new hypotheses. The experimental validation is done using the Brown corpus [FK79] in the context of a classification task on the genre of texts. For further experiments, we use the text corpus of the Project Gutenberg[8]. In particular, we test if the text analysis algorithm is able to distinguish books written by different authors and whether sections of books belong to the same or different books. Lastly, we use the Corpus Of Historical American English (COHA)[9] for an exploration of the linguistic changes in American English. Using texts published in the last 200 years, we explore to which extent the style of literature written in American English has changed during the last two centuries on a decade basis.

4.2.1 Brown corpus

The Standard Corpus of Present-Day American English [FK79] (Brown Corpus) is one of the first systematically organized text corpora. It was compiled by the linguists Francis and Kučera, both employed at the Brown University in the US state of Rhode Island, in the early 1960s. The corpus contains 1 million words from texts published in the United States of America in 1961. It is constituted by 500 text samples of around 2000 words each. In general, the samples characterize a wide range of styles and varieties of prose. Each text sample starts at a random point in the source article and continues to the first sentence following the first 2000 words. In very few cases, some samples contain fewer words. Furthermore, the 500 samples are categorized into 15 different genres. The distribution of text genres aims at being representative of the genres of texts published in 1961. In particular, the genres considered are: belles lettres (biographies and memoirs), adventure, fiction, mystery, romance, science fiction, government (government documents and industry reports), hobbies, humor, learned (scientific articles), lore, editorials, news, reviews and religious texts.

Some genres can be considered as subgenres of a more general category. In particular, we group adventure, fiction, mystery, romance and science fiction into a general fiction genre and editorial, news and reviews into the genre of press. We decided to group these

[8] https://www.gutenberg.org/

[9] https://www.english-corpora.org/coha/

genres for this classification task, because it yields to bigger text classes. This enables us to use a bigger training set while keeping the classes balanced by undersampling those containing more observations. In particular, we chose to use the 5 most frequent classes, selecting 48 texts of each. This results in the largest possible subset with at least 5 different classes and balanced class labels. The 5 biggest classes are belles lettres, fiction, learned, lore and press. Figure 4.1 shows the class frequencies of the grouped classes. The red line denotes the cut-off point. When dealing with bigger classes, we randomly sample 48 of their texts.

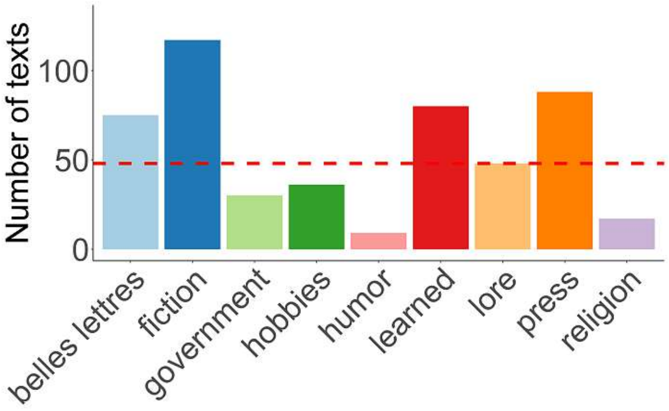


Figure 4.1: Histogram of the distribution of the grouped classes from the Brown corpus.

The classification experiment on the genres in the Brown corpus is set up as follows: first, we sample 48 texts from the 5 biggest genres (belles lettres, fiction, learned, lore and press). Next, we compute the pairwise text similarity between all pairs of selected texts. Lastly, we evaluate the classification performance using the methodology described in section 3.6.3.

4.2.2 Project Gutenberg corpus

The Project Gutenberg[10] is a free online library maintained on a voluntary basis. As of now, it contains over 60,000 licence-free e-books, which can be downloaded for free. Originally created by Hart in 1971, the Project Gutenberg is the oldest digital library existing. During the first years, literary texts were typed off manually before being uploaded to the servers. Later, book scanners and image processing software were used to digitalize the texts. Since the 1990s, the library is accessible from the public internet. From its start, the Project Gutenberg included mostly English texts, however during recent years, also numerous texts in other languages have been included. Furthermore, a significant proportion of Project Gutenberg texts was proofread by volunteers associated

[10] https://www.gutenberg.org/

with the Project Distributed Proofreaders[11], ensuring that the texts do not contain any syntactic or orthographic errors.

This community-based effort in collecting licence-free literature and making it accessible to the public is still ongoing, thus the corpus changes very frequently. New texts or literary artefacts are added, removed or modified on a daily basis. In order to create a static, consensual version of the corpus, which can be used for computational text analysis and other NLP purposes, Gerlach and Font-Clos published the Standardized Project Gutenberg Corpus (SPGC) [GFC20]. This corpus is described as an "open science approach to a curated version of the complete Project Gutenberg data containing more than 50,000 books" [GFC20]. In particular, the static corpus contains 55,905 books on 4 different levels of granularity:

- Raw data: it contain all downloaded books. Duplicate entries and entries not in UTF-8 encoding[12] are removed.
- Text data: raw data, but headers and boilerplate text are removed.
- Token data: text data, tokenized using the tokenizer from the natural language toolkit NLTK [LB02].
- Count data: list of words appearing in a text with the corresponding frequency.

Furthermore, each book in the static corpus is characterized by various attributes:

- id: book identifier, e.g. ’PG26’
- title: title of the book, e.g. ’Paradise lost’
- author: name of the author of the book, e.g. Milton, JohnMilton, John
- authoryearofbirth: year of birth of the author, e.g. ’1608’
- authoryearofdeath: year of death of the author, e.g. ’1674’
- language: code for the language of the book, e.g. ’[en]’
- downloads: number of downloads at the time of construction of the corpus, e.g. 737
- subjects: topics a book addresses. One book can feature multiple subjects, e.g. ’Adam (Biblical figure) – Poetry’, ’Eve (Biblical figure) – Poetry’, ’Fall of man – Poetry’, ’Bible. Genesis – History of Biblical events – Poetry’
- type: type of the item, e.g. ’text’

[11] https://www.pgdp.net/c/

[12] https://tools.ietf.org/html/rfc3629

In order to use the SPGC for validating the colexification-based text analysis method, we remove various groups of texts which are not suitable for our experiments: first, we select only entries in the corpus which represent a literary text from a known author. Entries with an empty author-field are removed. Additionally, we remove all texts with author entries as 'Various', 'Anonymous' or 'Unknown'. This step is necessary to perform the authorship attribution task. In Figure 4.2 the number of texts per author is displayed. Next, we filter for English language texts only. This is necessary since English is the reference language in the colexification network we are using and our analysis will only deal with English texts. Furthermore, we chose to select only texts belonging to the most frequent subject, fiction (see Figure 4.3 for the distribution of subjects in the corpus). This is done in order to have a conspicuous number of texts to analyse but also to control the number of variables which distinguish the texts we analyze. For example, in the authorship attribution task, it is more suitable to only use texts from one genre to ensure that the experiment is not corrupted by any unwanted, uncontrolled influences such a genre-specific literary style. In other words, in this experiment we want to be sure that the algorithm successfully recognizes authors and not the genre these authors wrote, therefore we select only authors that wrote fiction. Lastly, we remove texts which are too short, that is, we manually remove all texts containing less then 2000 tokens. In fact, the analysis of the Brown corpus was done by sampling 2000 tokens from a text, technique we want to apply also to the subsequent analyses. After the filtering procedure, the final text corpus consists of 39085 texts. Using the filtered static Project Gutenberg corpus, we perform two different text analysis tasks.

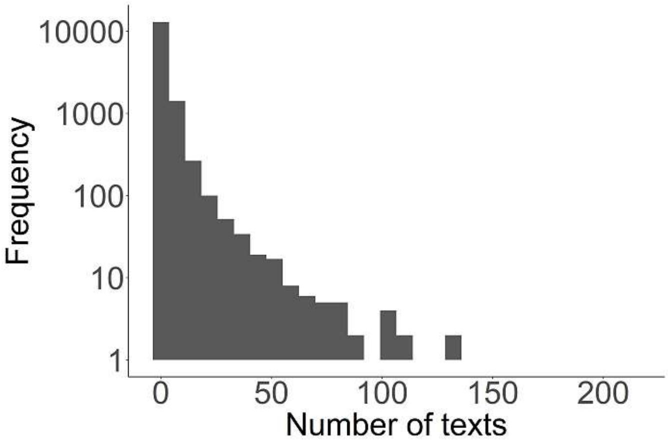


Figure 4.2: Distribution of the number of texts in the filtered SPGC by author. The frequency is displayed with a log scale on the y axis. On average, the corpus contain 2.62 texts per author, as indicated by the red dotted line.

Authorship attribution

In this experiment, we test if the colexification-based text analysis method is able to detect if two books are written by the same author or by two different authors. To

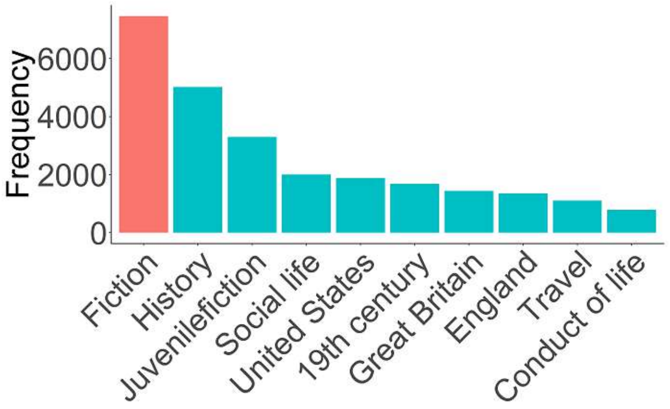


Figure 4.3: Texts in the filtered SPGC divided by the subject they deal with. Here, only the 10 most frequent subjects are plotted. Texts can belong to multiple subjects. As you can notice, the subjects indicated in the corpus do not always correspond to genres, as for example the 'United States' and 'England' labels. Fiction is the most frequent category, therefore we choose to analyse only texts belonging to it.

minimize the number of variables influencing the experiment, we chose to only analyse text in the filtered SPGC corresponding to the most frequent text subject, fiction. Many authors wrote texts that are classified as fiction. Table 4.6 lists the 15 most prolific fiction authors in the filtered corpus.

The experiment is set up as follows: we select 37 random texts written by the 15 most frequent fiction authors (which are listed in Table 4.6), resulting in a set of 555 texts. We decided to select 37 books because this choice maximizes the number of texts in the sample while keeping the sample balanced. Next, we compute the similarity between all pairwise combinations of books written by the same author. Thus, each text is paired with 36 texts written by the same author. Subsequently, for each book we select 36 random texts belonging to fiction but written by a different author and compute the corresponding text similarity values. This results in a total of 29970 text combinations. The text similarity algorithm is based on a random sample of 2000 words from each text. We therefore repeat each computation of text similarity 10 times with 10 different samples. We then evaluate the results using the ROC curve and AUC as explained in section 3.6.4.

Book excerpt recognition

The second experiment that deploys data from the Project Gutenberg corpus tests whether the text similarity measure is able to detect if two text samples origin from the same book or from different books. To test this, we consider books from the 5 most frequent fiction authors (see Table 4.6) in 5 independent experiments. We decided to perform the experiment for different authors in order to explore if the results are strongly

	Author	Number of texts
1	Jacobs, William Wymark	86
2	Trollope, Anthony	65
3	James, Henry	57
4	Fenn, George Manville	51
5	Hawthorne, Nathaniel	50
6	Le Queux, William	47
7	Crawford, Francis Marion	43
8	Haggard, Henry Rider	43
9	Bindloss, Harold	42
10	Doyle, Arthur Conan	42
11	James, George Payne Rainsford	42
12	Balzac, Honore de	41
13	Dickens, Charles	39
14	Yonge, Charlotte Mary	39
15	Howells, William Dean	37

Table 4.6: Table of the 15 most prolific authors in the fiction category belonging to the filtered SPGC.

dependent on a particular authors style, that is whether the results are generalizable. In each of the 5 repetitions of the experiment, we first compute all pairwise combinations of 2000-word excerpts of books from a particular author, including pairs consisting of two excerpts from the same book. Again, to account for the randomness introduced due to sampling 2000 words from each text, we repeat each text similarity computation 10 times using different seeds for the random samples. After the computations are completed, we evaluate the results using the ROC curve and AUC metric, as explained in section 3.6.4.

4.3 Historical analysis

The last text analysis exercises consist in the historical exploration of the colexification-based text similarity. In particular, we perform an analysis of the historical development of American English in fiction texts. The aim of this task is to provide insights into how the style and the range of topics covered during the last two centuries of literary fiction changed. In order to enable such an analysis, we use a linguistic resource of historical texts, the COHA corpus.

4.3.1 COHA corpus

The Corpus of Historical American English (COHA)[13] is one of the most frequently used text corpora in studies exploring the use of the English language over time. It was created

[13] https://www.english-corpora.org/coha/

by Davies, Professor of Corpus Linguistics at Brigham Young University (BYU)[14] and focuses of the American variety of English. Containing over 100.000 texts, COHA is the largest structured corpus of historical English. It contains over 1 billion words from texts written in American English between 1810 and 2009. Besides its identification number, each text in the corpus is also annotated with its year of publication, its author, title, length and one of four categories: 'fiction' ('FIC'), 'magazine' ('MAG'), 'news' ('NEWS') and 'non-fiction' ('NF'). The distribution of these categories is shown in Figure 4.4, while the distribution of the text length in number of words can be seen in Figure 4.5.

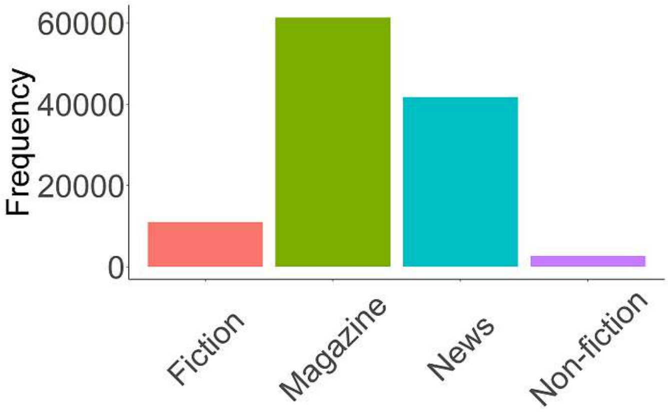


Figure 4.4: Distribution of the categories of texts belonging to the COHA corpus in number of texts. We decide to analyseonly texts belonging to fiction.

Similarly to the previous experiments, we have to pre-select texts from the original corpus. In particular, we filter the corpus following several criteria: first, we decide to include only texts belonging to the category fiction. This is done in order to minimize the amount of variables and stylistic influences to consider when comparing text similarities across time. Furthermore, we only consider texts which are composed of at least 2,000 words to avoid irregularities due to very short texts. After the filtering step, we obtain a subset of the corpus which includes 8,792 text. Furthermore, we decide aggregate texts according to their decade of origin, which we use as a temporal resolution for the following experiments. Figure 4.6 depicts the distribution of all texts in the COHA corpus according to their year of publication.

Intra-decade similarity

Using filtered corpus, we analyze how style varies with time by computing the similarity between texts belonging to the same decade (intra-decade experiment). First, we analyze how the intra-decade similarity changes with time. Therefore, we sample 40 texts from each decade and compute the similarity between all pairwise combinations of texts belonging to the same decade. For some of the earlier decades, less than 40 texts are

[14] https://www.byu.edu/

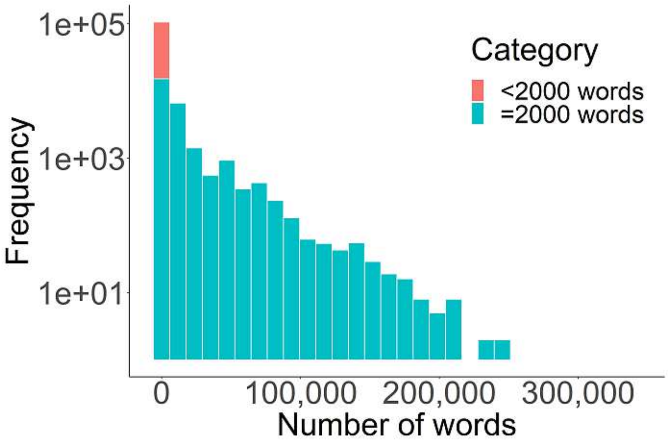


Figure 4.5: Distribution of the length of texts belonging to the COHA corpus in number of words. Most of the texts are longer than 2,000 words (colored red) while a smaller percentage, in blue, refers to texts that are shorter and have to be discarded for the task. The y axis shows the logarithm of the number of texts.

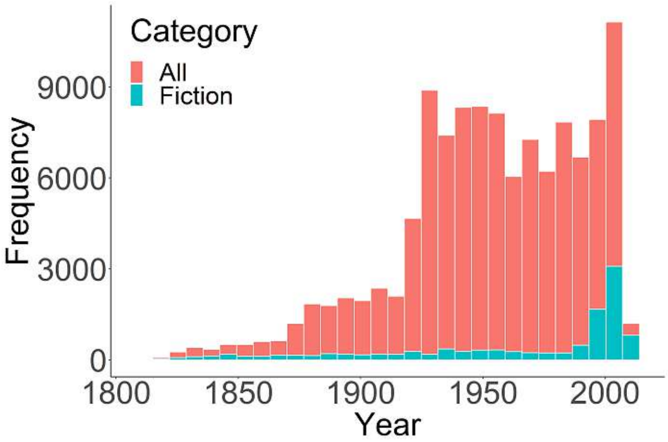


Figure 4.6: Distribution of number of texts in the COHA corpus according to their year of publication. The y axis represents the the number of texts. Fiction texts make up approximately 10 percent of the corpus. We decided to chose 40 fiction texts from each decade. The first two decades, 1810s and 1820s, are the only underrepresented ones.

available, as is shown in Figure 4.6. In such cases, we select the maximum amount of texts possible for this decade. As before, we repeat each combination 10 times with different random samples of 2,000 words. Thus, we ensure the robustness of the results. After computing all the similarity values, we evaluate the intra-decade similarity by aggregating the text similarity values belonging to the same decade and visually evaluating the resulting trends and similarity distributions.

Inter-decade similarity

In this final experiment, we analyze how and with which rate the American variety of English used in fictional texts changes stylistically over time (inter-decade). Similar to the previous analysis, we select 40 texts from each decade. Next, we compute the text similarity of all possible combinations of selected texts and repeat each computation with 10 different random samples. We aggregate the similarity values by combinations of decades. Lastly, we evaluate the performance of the model by visually inspecting the distribution of the results and by analyzing the ROC curves and corresponding AUC values.

CHAPTER 5

Results and discussion

The following chapter shows the results of the experiments discussed in chapter 4. These include word similarity prediction tasks, text classification tasks, as well as a practical application of the method. Moreover, this chapter discusses and interprets the achieved results, providing analyses and contextualizing the results with regards to related work in the field of computational text analysis.

5.1 Results

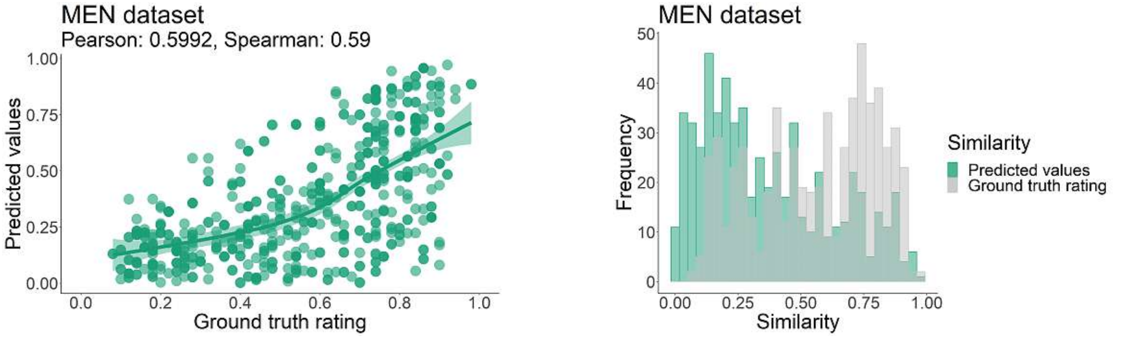
The results of the experiments described in sections 4.1, 4.2 and 4.3 are reported here. We use scatter-plots, boxplots, alluvial diagrams, as well as a carpet plot to visualize the results.

5.1.1 Word similarity

As described in section 4.1, the word similarity metric is evaluated using three datasets: MEN, SimLex and SimVerb, all described in section 4.1.1. These datasets collect word pairs together with their similarity or relatedness ratings. We use the MEN dataset to calibrate the similarity metric with regards to the dampening parameter β and several additional modifications. The other two datasets, SimLex and SimVerb, are then used for validation purposes only. In particular, we compare the results of our method with state-of-the-art algorithms on the last two databases. The following figures show the results achieved in said experiments. The figures are accompanied by a table 5.1, giving a condensed overview of the word similarity prediction results.

MEN dataset

The MEN dataset was used to calibrate the colexification-based word similarity metric. In Figure 5.1, we report the results obtained after calibrating the algorithm. In particular,



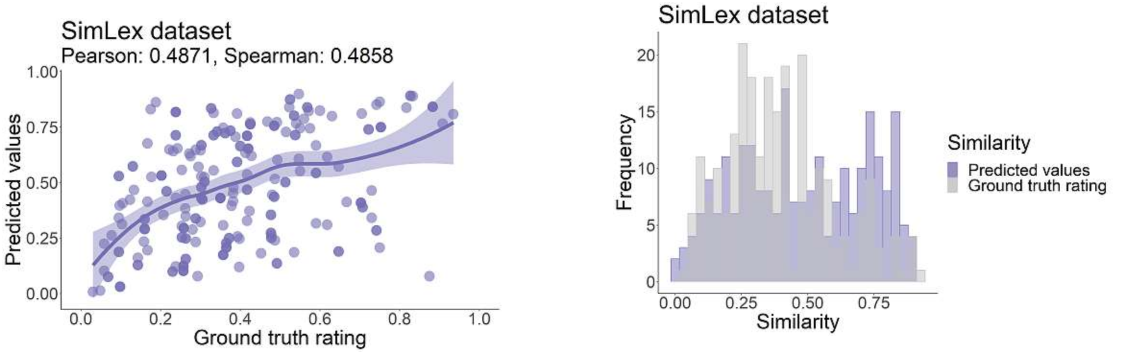
(a) Scatterplot of the predicted and ground truth word similarity ratings.

(b) Histogram of predicted and ground truth word similarity ratings.

Figure 5.1: Results of the word similarity prediction task on the MEN dataset.

in panel (a) the scatterplot of the predicted and ground truth values is reported, as well as the Pearson and Spearman correlation coefficients. We observe that the majority of the points in the scatterplot roughly lie around the 45 degree axis, whereby the proportion of deviation lower than said axis is larger then the proportion of points above it. This yields to a Pearson correlation coefficient of 0.60 (c.i.= [0.55,0.64], p<0.001) and a Spearman correlation coefficient of 0.59 (c.i.= [0.54,0.64], p<0.001) . From panel (b) we can see that the median of the distribution of predicted similarity values is significantly lower than the one of the ground truth values. Furthermore, the distribution of the predicted values is left-skewed while the distribution of the ground truth values in right-skewed. Therefore we observe a low overlap of the two distribution.

SimLex dataset



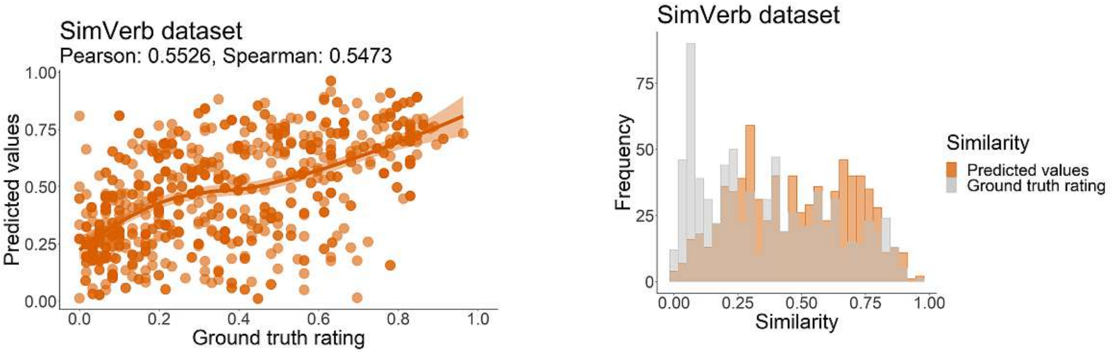
(a) Scatterplot of predicted and ground truth word similarity ratings.

(b) Histogram of the distribution of predicted and ground truth word similarity ratings.

Figure 5.2: Results of the word similarity prediction task on the SimLex dataset.

Figure 5.2 panel (a) reports the correlation coefficient between predicted and ground truth word similarity ratings according to the SimLex database. On this database, the colexification-based method reach a Pearson correlation of 0.49 (c.i.=[0.43,0.53], p<0.001) and a Spearman correlation of 0.49 (c.i.=[0.43,0.55], p<0.001). The histogram in panel (b) shows that the distribution of predicted similarity values has a higher median than the distribution of the ground truth values, which is the opposite trend respect to the results with the MEN dataset (Figure 5.1). In this case, the overlap of both distributions is higher than in the previous case. It has to be noted that the number of word pairs for which our algorithm was able to predict a similarity rating in the SimLex dataset is significantly smaller than in the previous dataset. The reason for this is the size of the respective datasets, since the relative coverage of words is very similar (see Table 5.1).

SimVerb dataset



(a) Scatterplot of the predicted and ground truth word similarity ratings.

(b) Histogram of the distribution of predicted and ground truth word similarity ratings.

Figure 5.3: Results of the word similarity prediction task on the SimVerb dataset.

Figure 5.3 reports the results on the SimVerb database. In panel (a) we can see the correlation between predicted and ground truth word similarity ratings, which corresponds to 0.55 with both Pearson (c.i.=[0.52,0.58], p<0.001) and Spearman (c.i.=[0.51,0.58], p<0.001) definitions. Moreover, the points tend to roughly concentrate around the 45 degree axis, whereby the standard deviation from it decreases with increasing distance from the coordinate system origin. The proportions of observations above and below the 45 degree axis seem to be of similar size. In addition, panel (b) shows the distribution of the predicted and ground truth similarity values of the SimVerb dataset. We note that the distribution of the predicted similarity values appears symmetric with a median close to 0.5. The distribution of the corresponding ground truth values, however, is very right-skewed with a significantly lower median.

Summary of the results of the word similarity prediction tasks

Table 5.1 provides an overview of the results presented in the previous figures. In particular, we report:

- n: number of unique words in the dataset;
- wp: number of word pairs in the dataset
- wp pred: number of word pairs in the dataset, for which a valid similarity; prediction can be computed using the word similarity metric;
- coverage: proportion in percentage of word pairs in the dataset for which a valid word similarity prediction can be computed;
- pearson: Pearson correlation coefficient and 95% confidence interval;
- spearman: Spearman correlation coefficient and 95% confidence interval.

	n	wp	wp pred	coverage	pearson	spearman
MEN	751	3,000	609	20%	0.60 [0.55,0.63]	0.59 [0.54,0.63]
SimLex	1,028	999	253	25%	0.48 [0.43,0.53]	0.48 [0.43,0.55]
SimVerb	827	3,500	779	22%	0.56 [0.52,0.58]	0.56 [0.51,0.58]

Table 5.1: Results of the word similarity prediction task on the three databases considered.

From Table 5.1, we can see that the SimLex dataset contains a much large number of unique words relative to the size of the dataset. Even if the number of word pairs and of unique words changes from dataset to dataset, the proportion of word pairs covered by the colexification-based similarity metric, however, is relatively stable. In fact, for each dataset a similarity assessment can be made for around 20 to 25 percent of the word pairs. Finally, the model achieves the best prediction performance (correlation coefficients of around 0.60) using the MEN dataset, which is to be expected since this dataset was used to calibrate the model parameters. The correlation coefficients of the other datasets are slightly lower, but still considerably high.

In order to put the colexification-based word similarity approach and its performance into perspective, we compare the previous results to results achieved by similar state-of-the-art models on the same databases. Such models, presented in 2020 in [RPPV20], use unsupervised graph word representations, in which each word is a node in a weighted graph and the distance between words is the shortest path distance between the corresponding nodes. In particular, the authors show that a graph-based approach of word embeddings better represents the structure of language than representations deploying vector spaces. As the approach presented in this publication is similar to our colexification-based approach proposed in this thesis, it is very interesting to compare the performances in predicting word similarity using the SimLex and SimVerb datasets, which is shown in table 5.2.

In Table 5.2 we can see that the colexification-based model achieves the best performance in both SimLex and SimVerb datasets. In particular, our method outperforms state-of-the-art models by approximately 250 percent on the SimLex datatset and approximately 80 percent on the SimVerb dataset.

Word embeddings	SimLex-999	SimVerb-3500
Euclidean GloVe	20.1	8.7
Poincaré GloVe	23.5	11.6
Graph GloVe	30.4	14.4
Colexification model	**47.6**	**55.6**

Table 5.2: Comparison of the word similarity prediction results of the colexification-based similarity model with the results of vector-based and graph-based word embeddings from [RPPV20]. The correlation coefficients are reported for each combination of model and word similarity database.

5.1.2 Author creativity prediction

The following Table 5.3 shows the results of the creativity prediction tasks with data from the forward flow experiment [GAC+19]. The first two entries report the Pearson and Spearman correlation coefficients between the similarity ratings estimated with the colexification method and the creativity score of the participants. The last two values correspond to the correlations of the forward flow estimates and the same creativity scores.

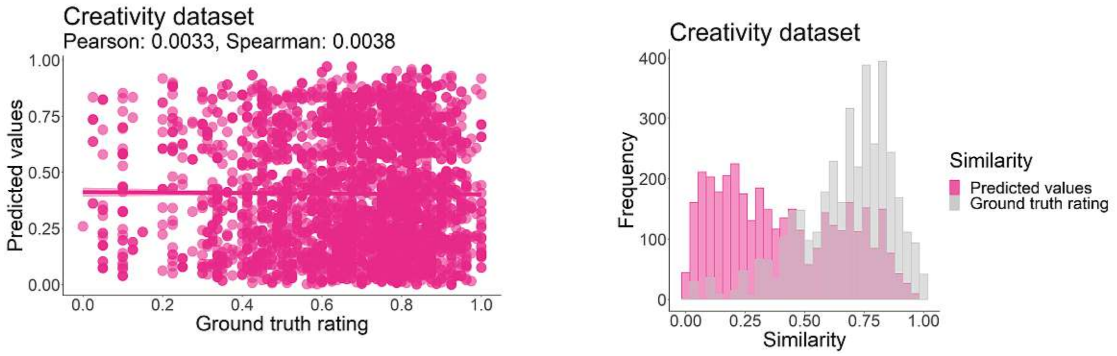
Pearson Colex	Spearman Colex	Pearson FF	Spearman FF
0.003 [-0.018,0.014]	0.004 [-0.016,0.020]	0.009 [-0.007,0.027]	0.001 [-0.015,0.018]

Table 5.3: Pearson and Spearman correlation coefficients and relative 95% confidence intervals on the creativity prediction task. The first two entries correspond to the colexification-based method, while the second two values refer to the performance of the forward flow (FF) estimate presented in [GAC+19].

The Table 5.3 shows that in both cases the correlation coefficients of the creativity prediction task are very low. The Pearson correlation of the colexification-based model is 0.003 (c.i.=[-0.018,0.014], p=0.207), the Spearman correlation coefficient achieved is equal to 0.004 (c.i.=[-0.016,0.020], p=0.243). The P-values show that the results are not statistically significant. The Table 5.3 also shows that the Forward flow measure does not manage to predict the author creativity with reliable accuracy either. However, the colexification-based model achieved a minimally higher correlation than the forward flow estimate. Moreover, the correlation values found are not significant, therefore we cannot draw conclusions from this experiment. Note that the correlations reported in the original forward flow paper [GAC+19] are definitely higher than the ones reported in this experiment. Surely the inclusion of more data and the inclusion of the full association chains would lead to better results for both methods. However, because of data availability issue, the analysis of the full data was not possible.

Figure 5.4 panel (a) shows the scatterplot of the predicted creativity values and ground truth creativity values achieved by the colexification-based model. No clear trend or correlation between predicted ratings and ground truth values can be observed.

Furthermore, from panel (b) we can see that the distributions of the predicted and ground truth values only overlap slightly.



(a) Scatterplot of the predicted and ground truth word similarity ratings. The correlation coefficients are close to 0 and not significative.

(b) Histogram of the distribution of predicted and ground truth word similarity ratings, which overlap only slightly.

Figure 5.4: Results of the word similarity prediction task on the forward flow dataset with the colexification-based word similarity model.

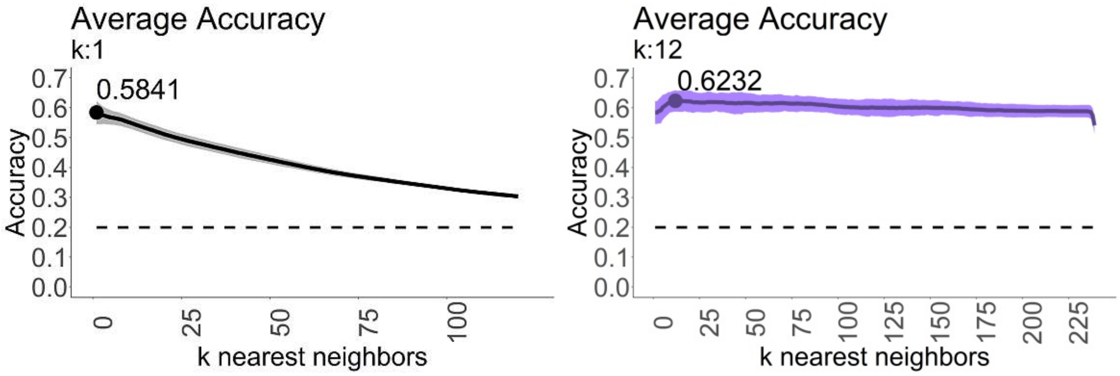
5.1.3 Text similarity

From the word similarity metric we define a text similarity algorithm based on it, which is described in section 3.5.3. We evaluate the performance of this algorithm deploying texts from the Brown corpus (described in section 4.2.1) and the Gutenberg corpus (introduced in section 4.2.2). In particular, we perform a genre classification task, and authorship attribution exercise and a book excerpt prediction. In the following sections, the results of these analyses are reported.

Genre classification using the Brown corpus

In the genre classification task we use texts from the Brown corpus and test if the text similarity metric is able to predict a the genre of a text. The prediction is based on the genre of the closest neighbors of the chosen text, as classified by the text similarity algorithm (see section 4.2.1 for more detailed information on the analysis). Figure 5.5 shows the results of the experiment deploying both the classic kNN and weighted kNN evaluation methods, as described in section 3.6.3.

As we can see in Figure 5.5 panel (a), the classification accuracy decreases monotonically with increasing number of neighbors k when taking into account the classic kNN prediction strategy. Furthermore, the width of the uncertainty band decreases the more data, i.e. the more neighbors, considered. The weighted kNN evaluation method is reported in the same Figure panel (b). It shows a less clear trend: the classification accuracy increases up until $k = 12$, where it hits its maximum. After that the accuracy slowly decreases with



(a) Classic kNN

(b) Weighted kNN

Figure 5.5: Performance in accuracy of the genre classification task evaluated using the classic kNN and weighted kNN prediction metric. The x axes represent the number of nearest neighbors considered. The light gray and light purple bands represent the uncertainty bands of two standard deviations given 120 different samples. The dashed line represents the baseline of a random classifier in this 5 class classification problem. Theoretically, this classifier would have accuracy equal to 0.20.

increasing parameter k. We can observe that the classic kNN method, with a maximum accuracy of 0.58 has a lower performance than the weighted kNN, which reaches an accuracy value of 0.62.

In order to have a better understanding of the performance of the method, we analyze which classes are the most misclassified by the algorithm. Figure 5.6 shows an alluvial diagram featuring the ground truth class labels and the predicted ones.

Analyzing Figure 5.6, we observe that the performance of the model is not constant across genres. In particular, books belonging to the genres fiction, learned and press tend to be classified with higher accuracy than theones belonging to belles lettres and lore. The rate of misclassification can be found in texts belonging to belles lettres and lore, which are classified as press. Furthermore, the genre lore is predicted the least amount of times whereas the genres learned and press are the most frequently predicted.

Next, we compare the results achieved by the novel text analysis method to a standard text analysis method serving as a baseline. The baseline method uses a tf-idf text representation and the cosine distance to compute the similarity between two texts. A more detailed explanation of the model can be found in section 3.5.4. Figure 5.7 show the results achieved by the baseline model when evaluating its performance using the classic and weighted kNN prediction metrics described in section 3.6.3.

Analyzing Figure 5.7, we can see similar trends compared to the colexification-based model. The classic kNN accuracy reaches its maximum for $k = 3$ and decreases monotonically afterwards. The accuracy of the weighted kNN approach increases significantly approximately until $k = 20$, and after that it stays relatively constant. The maximum in

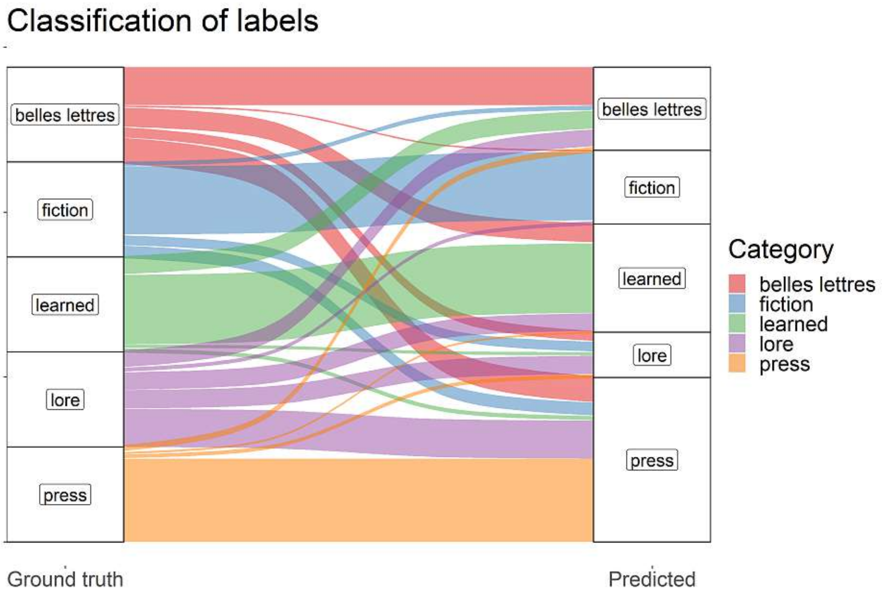
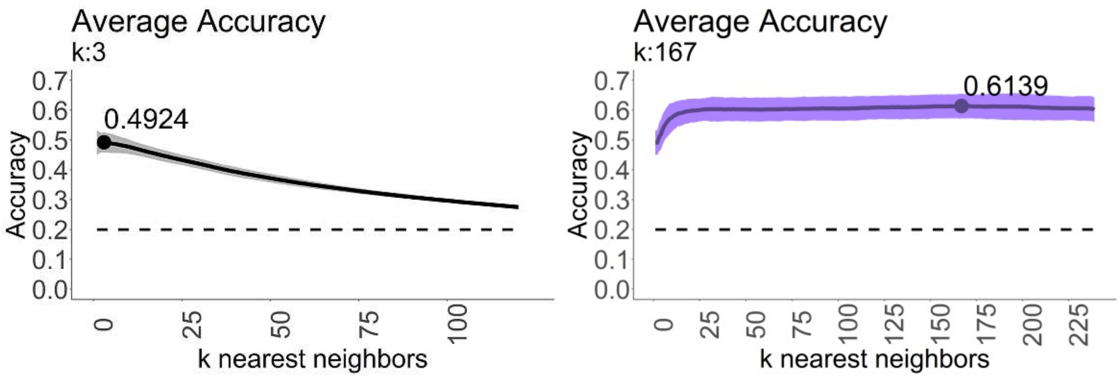


Figure 5.6: Alluvial diagram corresponding to the results of the genre classification task according to the classic kNN prediction strategy.



(a) Classic kNN

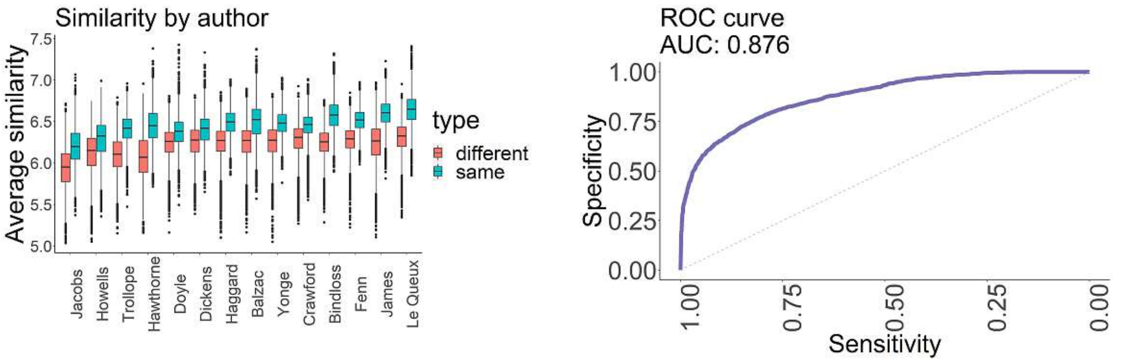
(b) Weighted kNN

Figure 5.7: Performance of the baseline model in the genre classification task evaluated using the classic and weighted kNN prediction metrics. The baseline method is based on a tf-idf scheme. The two standard deviation uncertainty bands are shown using light gray in panel (a) and purple in panel(b). The dashed line represents the theoretical results of a random classifier in this 5 class classification problem serving as a baseline. The baseline method has accuracy of 0.20.

accuracy is reached for a value of $k = 167$. Also in this case the weighted approach yields to better results than the classic one. Moreover, when comparing with the results of the colexification method, we find that our method yields to significantly higher results when taking into account the classic kNN approach. When taking into account the weighted kNN approach, the colexification method approaches an higher accuracy but in this case the difference between the two methods is not significant, since their uncertainty bands overlap.

Authorship attribution using the Project Gutenberg corpus

The authorship attribution experiment analyzes if the text similarity measure is able to distinguish between texts written by different fiction authors. The task is described in detail in section 4.2.2. The texts used in this task are sourced from the Project Gutenberg corpus (see section 4.2.2). In particular, we select texts from 15 fiction authors in order to keep the genre of the text constant and vary only the author variable. The following Figures report the results of the experiment. In particular, the ROC curves and the corresponding AUC measures are used to evaluate the results. Figure 5.8 shows the distributions of pairwise similarity values corresponding to books written by the same author (intra-author measure) and by different authors (inter-author measure), as well as the ROC curve for this task.



(a) Boxplot of the similarity value distributions of books written by the same author (blue) and by different authors (red).

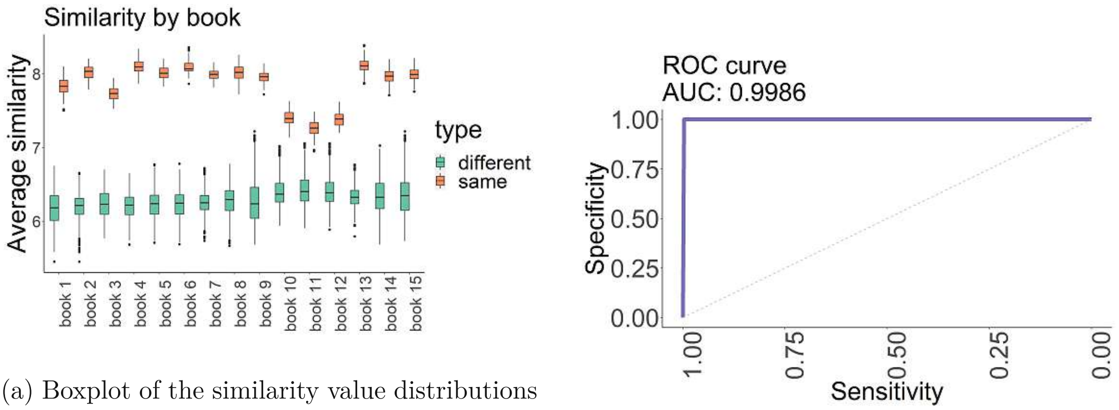
(b) ROC curve relative to the authorship attribution task. The corresponding AUC amounts to 0.88.

Figure 5.8: Results of the authorship attribution task

In Figure 5.8 panel (a) we can see that the median of the distribution of the similarities between books written by the same author is higher than the median corresponding to books by different authors. This is the case for each of the 15 authors featured in this experiment. Whereas some authors such as Le Queux, James and Hawthorne show a big difference between the two distributions, which overlap only in the tails of the distributions, for other authors, as for example Howells and Doyle the distributions overlap significantly. On an additional note, we observe that the variance of similarity values in both settings, intra- and inter-author, is high. Figure 5.8 panel (b) shows the ROC curve and its AUC corresponding to the results of the author classification task. The ROC curve shows a very smooth, concave shape, indicating a very balanced distribution of true positives with respect to the associated probability estimate. The corresponding AUC value is 0.88.

Book excerpt recognition using the Project Gutenberg corpus

Using a similar methodology to the one of the authorship attribution experiment, we analyze if the text similarity metric is able to distinguish if two book sections belong to the same book or to two different books. Said experiment is described in section 4.2.2 in more detail. We again use boxplots, as well as ROC curves and their AUC to evaluate the results. In order to show the robustness of the results, we perform the experiment considering the books written by each of the 5 most frequent fiction authors in the corpus. The results corresponding to one author are reported in Figure 5.9 and Table 5.4 summarizes the results for all 5 experiments.



(a) Boxplot of the similarity value distributions of the excerpt classification experiment. The orange boxplots represent the distribution of similarity values of excerpts from the same book and the green one correspond to excerpts coming from different books.

(b) ROC curve relative to the similarity distributions of the excerpt classification experiment. The relative AUC is nearly 1.

Figure 5.9: Book excerpt recognition with books written by W. W. Jacobs.

Figure 5.9 reports the results of the task regarding one of the authors taken into account in this experiment, namely W. W. Jacobs. The boxplot in panel (a) shows that the medians of the distributions of the similarity of book sections belonging to the same book are significantly higher than the corresponding medians of the similarity of book sections from different books. This is true for each book of each author in this experiment. Indeed, considering the ROC curve of each experiment, which is plotted in Figure 5.9 in the case of we W. W. Jacobs, and the relative AUCs, reported in Table 5.4 in the 5 cases, we can see that almost each predicted membership, i.e. the prediction of whether the two excerpts belong to the same or to a different book, is correct. Thus, the corresponding

Author	AUC
W. W. Jacobs	0.9986
A. Trollope	0.9996
H. James	0.9968
G. M. Fenn	0.9971
N. Hawthorne	1

Table 5.4: Summary table of the AUCs of the book excerpt recognition task with the 5 most prolific fiction authors in the corpus. 15 books written by each author were selected. Each combination of books was sampled 100 times, resulting in a total of 12,000 samples for each authors.

AUC values are equal or very close to 1 in each case. The variance of intra-book similarity varies slightly between authors: Hawthorne seems to write books covering the narrowest range of topics and using the most similar style, thus showing the smallest variance in the intra-book similarity distribution. On the other hand, the books written by James exhibit the greatest intra-book variance but still yield a very high AUC value.

5.1.4 Historical analysis

In addition to the classification tasks used for the validation of the text similarity measure, we perform a historical analysis of language used in fiction texts from the COHA corpus (see section 4.3.1). The experiment is described in detail in section 4.3. First, we explore the intra-decade similarity, i.e. the simialrity between texts written in the same decade, using boxplots, which are shown in Figure 5.10.

This Figure shows a slight but clear decreasing trend with increasing time. The similarity between books published in the earliest considered decades are the most similar to themselves. A clear increase of the rate of linguistic change seems to happen during the two most recent decades, the 1990s and 2000s. Another aspect worth noticing is the high variance of the similarity within each decades, shown by the long whiskers above and below the boxes.

Figure 5.11 depicts the change in similarity of texts with increasing time between their publication. In particular, each point in the figure denotes the average similarity between books written in the 1810s and every other decade until the 2000s, indicated by the value on the x-axis.

The Figure 5.11 shows a similar but stronger trend with respect to Figure 5.10. Indeed, the decreasing similarity between books published in the 1810s and more recent decades suggests that the similarity between two books is a function of time between their publications. Apart from a few small outliers, namely the 1860s and 1930s, the rate of change seems to be constant. Whereas Figure 5.11 shows the text similarity of each decade to the 1810s, the following Figure 5.12 includes all other combinations of decades as well.

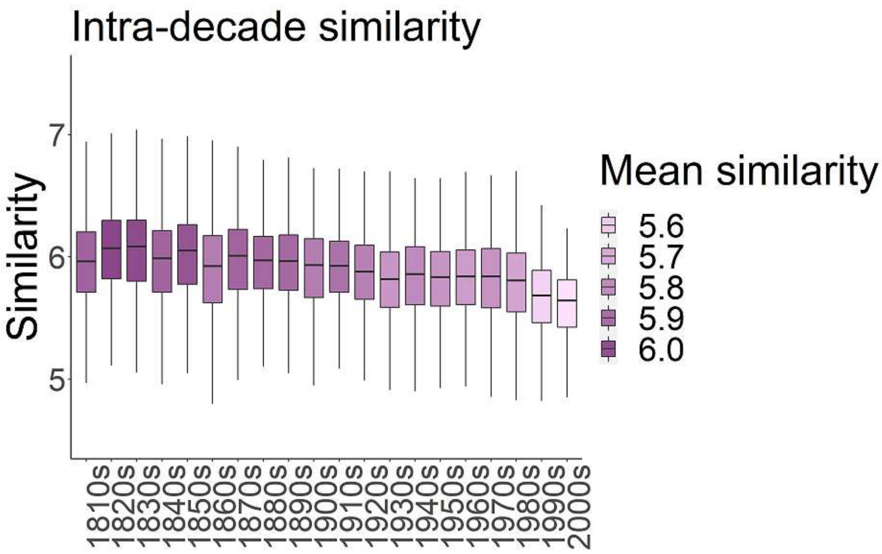


Figure 5.10: Boxplot of the similarity of fiction texts written in the same decade (intra-decade experiment). The boxplots are colored according to the mean of their similarity.

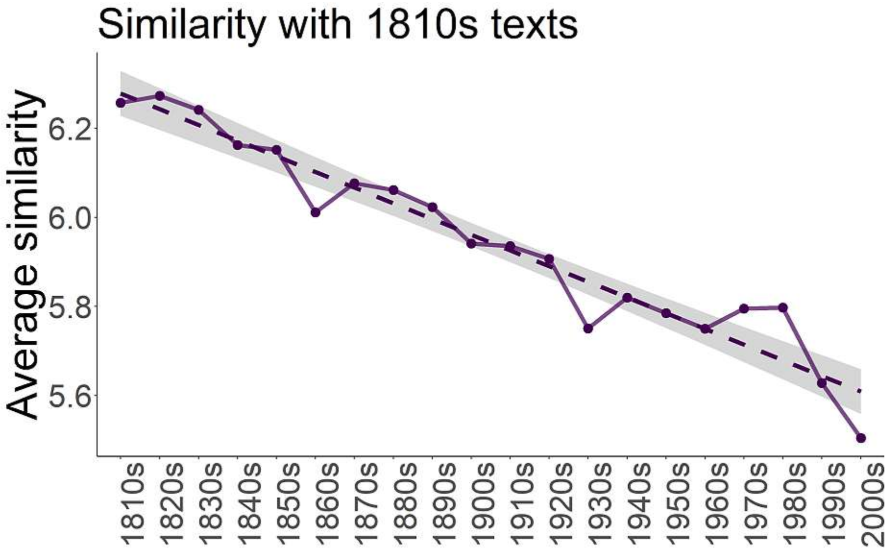


Figure 5.11: Average text similarity between books from the 1810s and texts published in all the other decades. The solid line represents the mean similarity, the dashed one is the fitted line showing the general trend and the grey area represents the 95% uncertainty band.

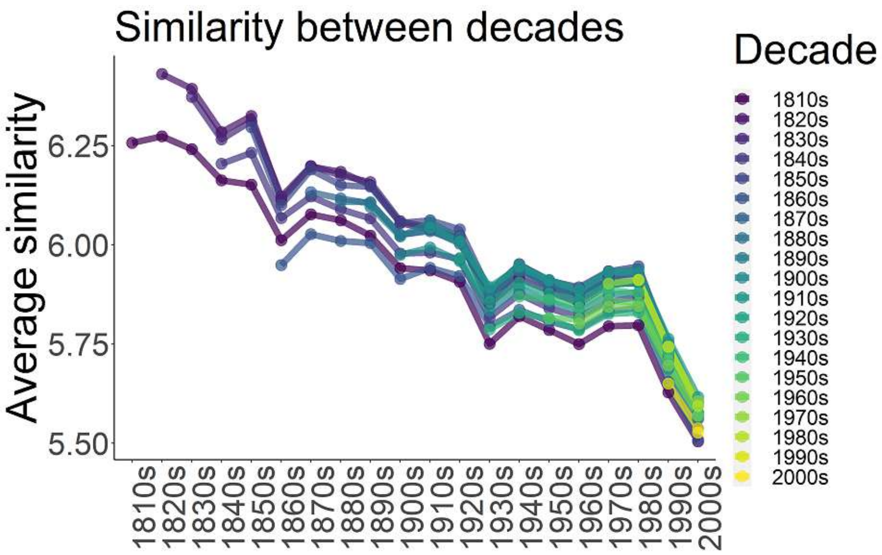


Figure 5.12: Average text similarity between texts written in different decades, as reported on the x axis. Lines are colored according to the reference decade.

Figure 5.12 shows that the main tendencies observed in Figure 5.11, which takes into account only the furthest decade, are present in all combinations of decades. In particular, the farther apart in time two decades are, the less similar books between them tend to be. Moreover, we see an increase in the rate of change when considering the last three decades, which reflects also the results of the intra-decade similarity reported in Figure 5.10.

In order to understand the relevant trends more in-depth, we fit a linear regression model to the data. This data includes the mean similarities between all combinations of decades explored in this experiment. Using a linear model, we try to explain the dependent variable relative to the mean similarity using the independent variables of the time difference between the publication of two books and the decade of publication of the first book. To be more precise, the linear function looks as follows:

$$\text{mean similarity} = a + b \cdot delta + c \cdot \text{decade1}$$

where *delta* stands for the difference in time between the publications of both books and *decade*1 represents the decade of the first (older) book as a numeric value. The coefficient a is the y intercept of the linear fit and b and c are the linear coefficients. Both linear coefficients b and c are negative and statistically significant. The y intercept a is equal to 12.51 and statistically significant as well. Table 5.5 presents the coefficients of the linear model and their properties.

	Coefficient	Estimate	Std. Error	t value	$Pr(>\|t\|)$	Signif. codes
(Intercept)	a	12.5159	0.2585	48.41	<2e-16	***
year1	b	-0.0034	0.0001	-25.13	<2e-16	***
delta	c	-0.0035	0.0001	-25.61	<2e-16	***

Table 5.5: Coefficients of linear model. The mean similarity per decade is the dependent variable. The time gap between two decades and the decade of origin of the first book are the independent variables. The standard error denotes the mean deviation that the coefficient estimates vary from the actual average value of our response variable. The t value describes the number of standard deviations the corresponding coefficient is away from 0 and the column $Pr(>|t|)$ denotes the corresponding the probability of observing a value equal or larger than t. The significance codes encode the column $Pr(>|t|)$ visually.

Finally, we also evaluate the results of the historical analysis with ROC curves and the corresponding AUC values. The following Figure 5.13 shows the AUC value of the binary classification results of each combination of decades. In particular, in this figure we represent the AUC relative to the decade discrimination exercise, that is, given two texts the task consists into deciding whether the two texts have been written in the same decade or not.

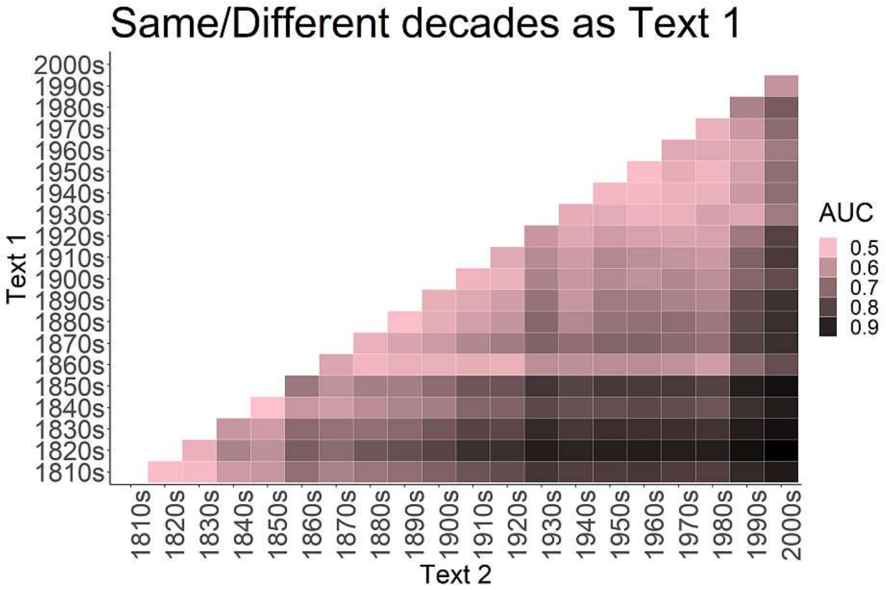


Figure 5.13: AUC of binary classification of the decade of writing of a book. The light color represent the performance of a random classifier and the darker the color is the better the algorithm is in distinguishing texts written in different decades.

Figure 5.13 shows a clear trend: the more time passes between the publication of two texts, the easier it is to predict if they were published in the same decade or in different decades. This trend can be identified visually analyzing the gradient from light to dark tones when going from left to right. This means that it is harder to distinguish books published in two subsequent decades and the longer the time gap is the easier the task is. However, we can observe some outliers which deviate from the general trend. In particular, the texts corresponding to the decade of the 1860s seem to be harder to distinguish from other texts. This can be seen by the fact that the corresponding line in Figure 5.13 is significantly lighter than its neighboring lines. A similar slight deviation from the respective trend can also be observed in Figure 5.10, 5.11 and 5.12. This seems to indicate that books from the 1860s are uncharacteristically dissimilar from other texts, which makes it more difficult for the binary classification model to predict if two books were written during the 1860s on the basis of their similarity value.

5.2 Discussion

In the following section we discuss the results shown in section 5.1. We start by interpreting the results of the word similarity tasks, which include the word similarity prediction using the MEN, SimLex and SimVerb datasets. While the MEN dataset was used for model calibration, the SimLex and SimVerb datasets were used only for the validation of the method we developed. We continue by interpreting the results of genre classification experiment with texts from the Brown corpus and the experiments using the Project Gutenberg corpus. Lastly, we focus on the interpretation of the historical analysis using the COHA corpus, analyzing what insights it can give us about the development of language used in American English fiction during the last 200 years.

5.2.1 Word similarity

We first discuss the results of the word similarity experiment reported in section 5.1.1, which aims at evaluating the performance of the word similarity measure. Since the word similarity algorithm forms the basis of the text similarity method, this first step is necessary to test the feasibility of our project. The word similarity measure takes into account the influences of all the paths between two words in the colexification network. In particular, the signal transmitted through the network edges are propagated such that nodes situated far apart in the network still influence the similarity computation to a certain degree. The degree of influence is determined by a dampening parameter, which we set to 0.8 in accordance to the Google pagerank algorithm [PBMW99], which uses a similar approach to computing the stationary visiting distribution in the respective network. Moreover, the calibration procedure using the MEN dataset has shown that this value of the dampening parameter is close to optimal. In addition, we think that opting for a strong propagation of the signal throughout the network is reasonable from a theoretical perspective. Indeed, natural language is a complex pattern with countless, far-reaching co-dependencies between words, concepts and groups thereof.

MEN dataset

Analyzing the results of the word similarity prediction task using the calibration dataset MEN (see section 5.1.1), we observe that the distributions of predicted similarity scores and ground truth similarity scores possess different characteristics: the median of the distribution of the predicted values is significantly lower than the median of the ground truth value distributions. We think that this happens because the MEN dataset includes all types of similarity and relatedness relationships between concepts, even those which might not be relevant for our metric. For example, the MEN dataset includes also whole-part relations, as for example the pair 'room' - 'window', which are not necessarily included in the colexification network. Furthermore, the high median of the ground truth distribution from the MEN dataset can be due to the collecting procedure of the ratings. Indeed, the scores were determined on a relative basis by ranking the word pairs on the basis of how often the participants deemed them to be the more similar word pair across a selection of such pairs in a binary decision. Therefore, the reason for the high median of the ground truth distribution might be that the most similar/related word pairs in the dataset tend to contain the most common words. Moreover, we consider only a subset of the database, namely only the pairs whose words are present in the colexification network. Indeed, for the other pairs it is not possible to compute their similarity with the colexification-based approach. This selection tends to exclude rare words, skewing the considered dataset towards more common words.

Furthermore, we see that the greatest source of prediction errors are high similarity ground truth observations predicted as being low in similarity by the model. We assume that is due to the incompleteness of the colexification network, which relates to the fact that it was constructed manually by linguistic experts and thus limited in its scope. A more extensive source of colexifications of the same quality, which could be used as a basis for the colexification-based word similarity metric, would most probably benefit the regression performance of the model greatly.

After calibrating the word similarity measure with the MEN dataset, we continued with the validation process on two new word similarity datasets, which is discussed in the following paragraphs.

SimLex dataset

As can be seen in section 5.1.1, the distributions of predicted and ground truth similarity values from the SimLex dataset show opposing characteristics compared to the MEN dataset discussed before. In general, in this case the two distributions are more balanced and thus overlap to a greater degree. The proportions of sources of errors for the prediction is balanced between overestimates and underestimates, indicating that the model manages to avoid being significantly biased towards one side of the spectrum. Moreover, while the distribution of the ground truth ratings is slightly skewed towards the left side (around a value of 0.25), the distribution of the predicted values is more uniform.

Overall, the correlation coefficients achieved in this task are significantly lower than those achieved with the MEN dataset. On one hand this is caused by the fact the MEN dataset was used to calibrate the model, which is therefore optimized to achieve the best results on said dataset. On the other hand, another reason for the drop in correlation might be that the SimLex dataset includes words belonging to various grammatical categories instead of nouns solely, like MEN. In particular, SimLex includes concrete and abstract adjectives, nouns and verbs. Because of this, it has been shown in [HRK15] that state-of-the-art methods in the field have difficulties in achieving correlation scores as high as in other datasets. A third reason for the decrease in prediction performance compared to the MEN dataset might be that the SimLex dataset explicitly focuses only on similarity and not on the more general concept of word relatedness. This is in opposition to the underlying colexification network, which includes various relations, including all types of relatedness. Moreover, the SimLex dataset is relatively small and our method manages to predict the ratings of only a small subset of these word pairs. Therefore, to make more definite statements on the performance of the method in this word similarity prediction experiment, we would either need more data or a more extensive colexification network yielding a greater coverage of the words appearing in SimLex.

SimVerb

Analyzing the results of the word similarity prediction task using the SimVerb dataset, which are reported in section 5.1.1, we observe the following: first, the SimVerb dataset is the largest dataset of the three and the word similarity measure seems to achieve a performance close to the one obtained on the calibration dataset MEN. Furthermore, the distribution of predicted values seems to be very balanced, while the distribution of corresponding ground truth values contains many more low-similarity word pairs than high similarity word pairs. The distribution of the ground truth ratings is skewed towards 0 probably because of the selection of word pairs in the dataset. The predicted values, however, are more evenly distributed, as happens also with the SimLex dataset. The reason of this might be twofold. First, we set a lower threshold on the similarity of words so that word pairs with a very low similarity value are not considered. Secondly, keeping into account the influences of all the nodes in the network up to a certain strength (depending on the dampening factor) might result in a more uniform similarity distribution.

Summary of results and comparison with state-of-the-art word embeddings

Next, we consider the summary of the results achieved in the three word similarity tasks, reported in Table 5.1. As expected, the similarity measure achieves the highest correlation coefficients on the MEN dataset, since it was used for calibrating the model. Moreover, we observe that the results on the SimVerb dataset are slightly higher than those obtained with the SimLex dataset. Our hypothesis for this evidence is that SimVerb collects word pairs rated on the basis of similarity and relatedness, similar as the principle of colexification itself, as defined in [Fra08]. SimLex, on the other hand, collects word pairs rated on the basis of similarity only, which is a more specific concept. Moreover, the SimVerb dataset contains only verbs. The results obtained on this dataset make us think that the colexification-based measure performs better when taking into account verbs.

The coverage of word pairs, i.e. the proportion of word pairs for which a prediction of similarity can be made, is relatively similar for all the datasets. The highest coverage is achieved with the SimLex dataset, which might be caused by the fact that it contains multiple grammatical classes of words, as the underlying colexification network does, opposed to the MEN dataset, which only contains nouns, and the SimVerb dataset, only containing verbs. Moreover, a maximal coverage of about 25% might seem relatively low. A significant reason for the low coverage of the three databases is the lower threshold introduced in in the modification of the similarity matrix (see section 3.4.5), which removed half of all similarity values in the similarity matrix. We chose to introduce this lower threshold at the cost of a significant reduction of coverage because the aim of this metric is not to cover the greatest range but to provide reliable word similarity predictions, which subsequently form the basis for the text similarity measure. Moreover, linguistic sources, as the colexification network is, rarely manage to have high coverage of a language. Therefore, only with the enhancement of the coverage of the colexification network more word pairs could be taken into account. Lastly, we find that the Pearson and Spearman correlation coefficients on all three datasets don't deviate significantly. This implies that the linear correlation of the results is similarly strong as their rank correlation.

In order to put the results achieved with the colexification-based word similarity measure into perspective, we compare them to other state-of-the-art methodologies on the validation datasets SimLex and SimVerb. The state-of-the-art methods chosen consist in different versions of word embeddings. In particular, we consider the results of an Euclidean, Poincaré and graph word embeddings as reported in [RPPV20]. The comparison of the results of the different methods is reported in Table 5.2. In that Table we can see that the colexification-based model clearly outperforms the other word embeddings, validating the importance of our results. However, as already stated, it is important to note that the colexification-based model does not cover the full ground truth datasets. Only 20 to 25 percent of all ratings can be predicted, thus impacting the performance evaluation. No information about the corresponding coverage of the word embeddings models is reported in [RPPV20], thus further analysis is required. We therefore conclude the following: the colexification-based measure outperforms state-of-the-art methods significantly, but probably on a different, smaller subset of the dataset. Thus, the results are not perfectly comparable. Nonetheless, the results in the word similarity prediction task are very encouraging and we deem the model to be suitable as a basis for the text similarity measure.

Author creativity prediction

In the creativity prediction task we used data from a previously published study [GAC+19] and tried to predict the creativity of a participant from the similarity score of the word association chain produced by said participant. The results shown in Table 5.3 suggest that neither of the word similarity measures, colexification-based metric and forward flow distance, is able to predict the creativity of the participants of the original experiment. Indeed, even if the correlation of the creativity score with the similarity estimates is higher in the case of the colexification-based model than in the case of the forward flow estimate, they are very low and not significative. One very probable reason for this is the bad quality of the available data, which does not provide creativity values of the same granularity as our predicted values. Moreover, the availability of only a small subset of the data used in the original study contributed to the achievement of not satisfying results. Further analysis is needed to explore the hypothesis that colexification-based measures can predict the creativity of a participant in the study.

Originally, we planned to explore whether the colexification-based word similarity measure can be used as a predictor of creativity. Inspired by the paper presenting the concept of forward flow [GAC$^+$19], we wanted to analyze the rate of similarity change between consecutive words in association chains and compare the mean similarity across a chain with the creativity ratings of the author of the word chain. The underlying assumption was that the more creative a person is, the more this person tends to associate dissimilar concepts, whereas the association chains created by less creative people tend to connect concepts which are more similar to each other. The forward flow data was necessary for this experiment because they collected creativity ratings of all the participants by deploying psychologically relevant tests.

Unfortunately, the authors of [GAC$^+$19] only provided a small subset of the original data used in the analysis. Indeed, out of the 5 studies based on association chains reported in the paper, we were only able to access the original association chain data of two of them. Thus, we could only compare the association chains of a group of actors, a group of Amazon Mechanical Turk workers and the ones corresponding to a representative sample of Americans. Subsequently, the small size of the available dataset led us to frame the task differently: we decided to shift our focus from the level of association chains to the level of pairs of consecutive words in order to increase the sample size. However, each creativity value associated with a word pair was retrieved from data with a different granularity. In particular, we could give a different similarity score to each word pair but the creativity was measured on the author level. Our starting hypothesis was that the mean of the similarity ratings of words in association chains written by creative people would be higher than the one for chains created by people with a lower creativity score. However, with the new experiment set up we could not consider the mean similarity values over the full association chain but only the single similarity scores. Even if it is acceptable to think that the mean creativity score across the association chain would be related to the creativity of the author, the creativity of the single pair of words might show a less clear pattern. Creative people might from time to time give a lower than average answer to the task. Therefore, the decision of considering word pairs added noise to the measure.

The results using the available data show that the colexification-based measure is not able to predict the creativity of the author of a word pair in the corresponding association chain. However, it is impossible to make any definite conclusions on the basis of this adjusted experiment. It is important to note that the forward flow measure was not able to predict the author creativity in this adjusted experiment either. Contrary to this, the original paper [GAC+19] has shown that the forward flow measure was indeed able to predict author creativity on the association chain level of observations (Spearman correlation of 0.19, $p<0.001$). We assume that the main reason for this is the different granularity of the experiments word pairs and the corresponding ground truth creativity values. While it is difficult to predict the potential results that would have been obtained using the full association chain dataset, we think that the colexification-based similarity measure would be able to predict the author creativity with reasonable accuracy.

5.2.2 Text similarity

On the basis of the word similarity metric, which reached promising results in the word similarity prediction task, we built a text similarity measure, and validate it in several classification tasks. Moreover, we perform some exploration tasks to inspect the insights that our method can give. The results of these experiments are reported in section 5.1.3.

Genre classification

First, we validate the performance of the model with a genre prediction classification task based on texts taken from the Brown corpus. The results are presented in section 5.1.3. Figure 5.5 shows the achieved classification accuracy using two different way of evaluation: a classic kNN approach and an advanced, weighted version thereof. Both are described in section 3.6.3.

Regarding the classic kNN approach, which is shown in Figure 5.5 panel (a), we observe that the classification accuracy decreases monotonically. The more neighbors we include, i.e. the higher the parameter k is, the lower accuracy values are achieved. This characteristic might origin from the fact that the variance of the predicted similarity scores is relatively low: 90 percent of the similarity score between books are between 5.12 and 6.85. The low variance of similarity scores makes it more difficult for the primitive, majority-based kNN evaluation method to distinguish between book genres. Moreover, the higher the parameter k is, the more noise we include in the evaluation measure. In fact, in this task we consider only 48 texts per genre. Therefore, even supposing to have a perfect classification algorithm that ranks the texts belonging to the same genre as the most similar, when considering more than 48 texts we will be introducing additional texts belonging to a different genre, that is we will be including noise in the measurement. Furthermore, we observe that the standard deviation of the classification accuracy decreases with increasing parameter k, since more data is included.

The idea behind the weighted kNN approach was to create a modified version of the classic kNN method, which does not yield a well-behaved solution, instead predicts the genre of an observation only considering the genre of its closest neighbors. The advanced approach, named weighted kNN, is not just majority based, but it also considers the distance (i.e. similarity) of the neighbors of an observation, as well as the frequency of observations of each genre among the neighbors. In Figure 5.5 panel (b) the results according to this second approach are reported. We observe that the advanced method performs better than the classic kNN approach, achieving a maximum accuracy 4% higher than the classic approach. The best results are achieved when considering $k = 12$. Moreover, the evidence that the maximum accuracy of the classic kNN approach is reached when $k = 1$, i.e. when taking into account only the genre of the closest text in similarity, indicates one reason why the weighted kNN method achieves better results. After the peak at $k = 12$, the classification performance decreases slowly but constantly. We think that the reason for this is that the inclusion of more neighbors leads to more noise, as happens also in the classic kNN approach.

Next, we consider the alluvial plot in Figure 5.6. We note that observations belonging to the genres fiction, learned (i.e. academic texts) and press are predicted with a higher accuracy than the ones belonging to the genres belles lettres (i.e. memoirs) and lore. Our explanation is that, while the first class of genres are very stylistically distinct, the genres in the second group have much fuzzier, unclear boundaries in terms of style and content. Moreover, the genres belles lettres and lore seem to be less well defined in the first place, adding another source of uncertainty and thus also a misclassification potential. It has to be noted that this is merely an assumption, which might be confirmed with further analyses, for example by conducting a benchmark test with human participants. We also observe that the genre learned is clearly the best predicted one, which is most probably due to the very precise, distinct style and topics of academic articles, a criterion which clearly distinguishes it from other text genres. Moreover, the alluvial plot shows that the predicted classes are imbalanced, with press being the most frequently predicted class and lore the least. This contributes to the evidence that press is one of the genres on which the model works best and lore is one on which the algorithm achieves the worst results. Thus, by setting the output of the model to balanced classes the performance on the genre classification task might be improved.

Also in this case, we put the colexification-based model into perspective with regards to other NLP approaches. This is done by comparing the results of our method with the results achieved by a baseline model, which is described in section 5.7. The baseline model, which uses a tf-idf text representation and the cosine similarity metric to estimate the similarity between two text vectors, serves as a basis for comparing the classification performance of our colexification-based model to standard NLP approaches. We analyze both evaluation approaches: classic and weighted kNN and report the results in Figure 5.7. The baseline model shows a very similar trend to the colexification-based method when taking into account both kNN approaches. In the case of the classic kNN, the classification accuracy is monotonically decreasing with a maximum performance reached with $k = 3$. In terms of absolute performance, the baseline model performs significantly worse than the colexification-based model in this setting. A general reason for this could be that the tf-idf text representation incorporates only statistical information about the frequency of words while the colexification-based model considers in addition the semantic content of the text. When taking into account the weighted kNN approach, both models show a very similar, well-behaved trend: the accuracy increases up until a value of k around 20, where it reaches a maximum. The maximum accuracy is achieved at $k = 167$ with a value of 0.61. In this case, the performance of the colexification-based model is higher than the one of the baseline, but not significantly. In other words, the results of the two models are statistically indistinguishable. Contrary to the colexification-based model, however, the baseline models performance does not decrease significantly from that point on. Instead, it stays relatively constant.

In general, the weighted kNN approach proved to be the one achieving the best results when predicting the genre of books. The colexification-based text similarity model predicts book genres with better or indistinguishable performance to the one of the baseline NLP model, even though both solve the problem differently. We suspect that both models base their similarity evaluations on different characteristics of the texts because they deploy different approaches: the colexification-based model uses the semantic analysis of words, while the tf-idf scheme deploys the statistical analysis of frequency of words. Thus, the combination of the models might lead to better results as both models might complement each other. Furthermore, as already mentioned, it would of great interest to quantify the performance of human raters in the genre prediction task. This would allow the comparison of the results of NLP models with the inter-rater-agreement scores of the human raters. This experiment is also needed to provide a support for our hypothesis that some genres (the most misclassified) have fuzzier boundaries and that this phenomenon introduces a significant source of difficulty for humans as well. Thus, the inter-rater-agreement score could be seen as a theoretical upper bound for prediction accuracy achievable by NLP models on this task.

Authorship attribution

The results of the second validation experiment, a binary classification based on the Gutenberg corpus, are presented in section 5.1.3. Figure 5.8 panel (a) shows the distribution of similarity scores for books written by the same or by different authors for each of the 15 most prolific fiction authors in the corpus. We observe that books written by the same author are, on average, more similar than books written by different authors. In particular, some authors seem to have a more distinct style of writing than others, as can be seen by the varying overlap between the inter-author (similarity between books written by different authors) and intra-author (similarity of books written by the same author) boxes corresponding to each author. While this is an interesting insight, it is difficult to support with further evidence since it is hard to quantify the distinctness of an author's style. In Figure 5.8 panel (b) the ROC curve and the relative AUC for the authorship attribution task is reported. The AUC amounts to 0.88, which confirms the hypothesis that the colexification-based text similarity model is able to distinguish the authorship of a book. In general, we consider this validation experiment as a success, showing that the colexification-based text similarity metric is able to reliably predict if two books were written by the same or by different authors.

Book classification

In this experiment, whose results are reported in section 5.1.3, we analyze if the method is able to distinguish whether two book excerpts belong to the same or to different texts. Such texts are selected among all books published by the same author, in order not to contaminate the task with an authorship attribution part. The hypothesis underlying the experiment, similarly to the one of the previous task, is that two book excerpts stemming from the same book tend to be more similar to each other than two excerpts from different books. We assume that this difference is even clearer than in the authorship experiment since the topics covered in a single book tend to have a narrower range than the topics an author covers among his/her whole bibliography.

As can be seen in Figure 5.9 and in the summary Table 5.4, the results of this experiment confirm our hypothesis. The boxplot in Figure 5.9 panel (a) shows that there is a very clear difference between the distribution of the similarity of texts belonging to the same book and the one of the similarity of texts belonging to different books. This trend prevails among all of the 5 authors featured, as reported in Table 5.4. Furthermore, some books seem to have a more distinct style or a more distinct range of topics covered as opposed to others, which can be seen analyzing the boxes in the boxplot. The corresponding ROC curves show a very similar picture: for each of the 5 experiments, the model is able to predict if two book excerpts are from the same or from different books with almost perfect accuracy. We assume that this is the case since most books revolve around a few selected topics throughout the whole book, feature that is identified and deployed by our model.

The book excerpt classification task yields to very high AUC values, always approaching 1 when considering 5 different authors. Indeed, we consider the set up of the task to have contributed to the simplicity of the task, indeed it is constituted by a binary classification exercise in which a random classification algorithm would be right 50% of the times. Moreover, our algorithm matches words from the two excerpts as its first step, which might have had a decisive role in the successful outcomes. In fact, it is very easy to tell whether two excerpts belong to the same book when the names of the characters and the setting match.

5.2.3 Historical analysis

After validating the text similarity measure using the three experiments previously described, we prove that the model can also be applied to a more exploratory task, a historical analysis of how the language used in American English literary fiction developed during the last 200 years. The results of this analysis can be found in section 5.1.4.

Figure 5.10 shows that the median similarity between books published in the same decade tends to slightly decrease with time, which indicates that the range of topics covered by literary fiction become broader over time. Taking a closer look at the boxplot shown in Figure 5.10, we observe the following trends: the intra-decade similarity, i.e. the similarity between texts written in the same decade, seems to be relatively constant when taking into account texts published from the 1810s to the 1870s, followed by a steady drop between the 1880s and the 1930s. After the 1930s, the median intra-decade similarity tends to remain stable again until the 1970s. During the last 3 decades of the analysis, the decrease in similarity accelerates. This development might be connected to the introduction of internet and other mass media during this time. This evidence can be linked to insights from [LSMHL19]. In particular, the authors consider collective attention in online and offline environments, confirming that the rate at which content becomes prominent and successively lose popularity has accelerated. One main driver for this evidence is the "increasing production and consumption of content, which results in a more rapid exhaustion of limited attention resources" [LSMHL19].

With the results reported in Figures 5.11 and 5.12 we explore trends of change of the similarity between books from different decades. A general trend we observe consists in the decrease in similarity of texts the higher the difference in time between the release of two books is. This seems plausible since each epoch is characterized by its own problems and most relevant topics, which fiction tend to address. In Figure 5.11 we analyze the difference in similarity between books published in the 1810s and books written in more recent decades. We observe that the rate of change seems to be relatively constant until the 1960s, at which point it seems to stagnate for two decades. After that, however, the similarity decreases sharply during the last two decades of our analysis, the 1990s and 2000s. Moreover, we find two outliers to the trend, namely the 1860s and 1930s, which are more dissimilar than what expected. Figure 5.12 confirms that this trend is observable also when taking into account all combinations of decades. Indeed, the two outliers are present and the rate of change sharply increases during the 3 most recent decades, confirming the insights on the accelerating dynamics of collective attention presented in [LSMHL19]. A similar exploration of the similarity between books published in different decades was performed in the paper presenting the standardized Gutenberg corpus [GFC20]. The conclusions presented are in agreement with our insights: with increasing time difference between the publication of two texts, the distance between them increases continuously. Furthermore, they also find that the rate of change increases significantly during the most recent decades.

Analyzing the linear model presented in 5.1.4 leads us to the following insights: both coefficients b and c, corresponding to the time difference between two books and the year of origin of the older book respectively, are statistically significant, which can be confirmed by inspecting the coefficient t values and corresponding probabilities $Pr(>|t|)$ in Table 5.5. Thus, we can confirm that two general trends impact the similarity between two books in a significant manner. On one hand, the similarity is dependent on the amount of time two decades are apart. Additionally, however, a secondary negative trend over time can also be observed. This means that pairs of books with the same time difference are less similar the later they have been written. For example, the similarity between a book written in 1810 and one written in 1850 tends to be higher than the similarity between two books published in 1910 and 1950.

Figure 5.13 shows the mathematical evaluation of the binary classification task aiming at predicting if two text were written in the same decade or not. On the basis of the darkness of the respective square, we can evaluate the AUC relative to the prediction. The hypothesis behind this experiment is that the similarity between two books decreases with increasing time between their decades of publication. Some general trends are observable. Firstly, in general the matrix entries becomes darker the farther on the bottom right the square is. This corresponds to an increase in classification performance with an increase in time between the origin of two books, which confirms our hypothesis. Furthermore, we observe that this trend is present with varying degree among all decades, as can be seen analyzing each row and column. Secondly, we observe that the row corresponding to the 1860s is much lighter than its neighboring rows. This means that the model encounters more difficulties in the task when one of the two input texts have been written in the 1860s. While the reason for this evidence is not clear, a similar deviation from the trend can be seen in the other figures as well. Indeed, as previously observed in Figure 5.10, the intra-decade similarity when taking into account the 1860s seems to be slightly lower then the trend suggests. Moreover, Figure 5.11 shows again a deviation from the general trend when taking into account the 1860s: the average similarity between books from the 1810s and the 1860s even falls below the uncertainty bands, as it happens also with the 1930s. Indeed, also the 1930s seem to portray a similar deviation from the general trend. This can be seen by comparing the darkness in Figure 5.13 of the column corresponding to that decade to the neighboring matrix columns. The cause of both these deviations is not clear but it might be due to the text sampling in the dataset.

CHAPTER 6

Conclusion

The following chapter includes a summary of the findings and insights found as results of this work. Furthermore, in this chapter we put them into perspective with regards to other methods in the field of computational text analysis and describe possible ideas for future work.

6.1 Summary

The main aim of this work was to construct a colexification-based text similarity measure, validate it on several experiments and thus confirm the hypothesis that colexification occurrences can be used in the field of text analysis. In particular, our work is based on the hypothesis that colexification encodes meaning similarity, which lacks validation at scale but has been taken for granted in various studies and applications of the idea of colexification. Following from this hypothesis, in this work we prove that colexifications can be applied to NLP. In particular, we construct a word similarity measure based on the Clics^3 database, a collection of colexification instances manually constructed by experts. Using this bottom-up approach, we made sure that the method we constructed is fully transparent and interpretable - a characteristic which many of the most used text analysis method in NLP miss. We calibrate and validate the measure using various standard NLP tasks, confirming that the measure can predict the semantic similarity between words with reasonable accuracy. Once developed a meaningful word similarity tool, we deploy it for the construction of a text analysis method.

The main product of this work is a colexification-based text analysis tool which computes the semantic similarity between two texts. Based on the word similarity measure build deploying the colexification network, we construct the text similarity method taking inspiration from a state-of-the-art text similarity measure [II08]. By validating the model using several text classification tasks framed around the Brown corpus and the Gutenberg corpus, two of the most prominent text corpora in computational linguistics, we confirm that the principle of colexification can indeed be used as a basis for knowledge-based text analysis. We show that the developed text similarity measure can be used to predict the genre of a book and distinguish whether two text have been written by the same author and if the are two excerpts of the same book.

We compare the results of the colexification-based similarity measures with results achieved by other state-of-the-art measures from the field of NLP. As a consequence, we can confidently claim that our approach achieves competitive levels of performance, and in some cases it outperforms machine learning approaches. In particular, we show that the word similarity measure significantly outperforms similar methods based on traditional word embeddings and the graph-representation thereof on the SimLex-999 and SimVerb-3500 datasets. While the coverage of our method might not be as competitive as the one of these methods, the results achieved by our measure are encouraging at least. Furthermore, we compare the colexification-based text similarity measure to a baseline model using standard text representation methods from the field of NLP, showing that both methods performances lie within the same confidence intervals and that in some cases the colexification-based approach reaches better results. Since both models approach the problem from completely different sides, namely from a semantic versus a purely statistical perspective, we assume that they make use of different text characteristics on which to base their predictions. Regarding future work on this topic, it would be of great interest to combine both models and test if significantly better prediction results can be achieved. Moreover, one of the advantages of the proposed text similarity model lies in the fact that it relies on the analysis of a very small subset of the input text. Indeed, contrary to other machine learning approaches that require massive training datasets, our approach does not require training and considers only 2,000 words from the input test. Our analyses show that, even with this small amount of data, the method is reliable and competitive to other state-of-the-art techniques. Not needing a training dataset, the method is also domain independent, thus suitable for the analysis of texts belonging to various domains.

Finally, we apply the validated text similarity measure to an analysis of the historical development of language used in American English literary fiction. With this experiment, we aimed at showing that the proposed method can be used to answer practical questions in the field of linguistics and give insights into how language and writing evolved during the past 200 years. The experiment, which is based on the historical corpus COHA, shows that the time gap between the release of two texts influences their semantic similarity. In particular, the text similarity tends to linearly decrease with an increasing difference in time. Interestingly, from the 1980s this decrease accelerates significantly, evidence that we hypothesize is caused by an acceleration of the flux of cultural content and link to the change in patterns of collective attention, which have been studied in a related work in computational social science [LSMHL19]. In general, our analyses confirm that the principle of colexification provides a valid basis for a knowledge-based measure of semantic similarity, which can be applied to different questions about human language.

6.2 Outlook

The outcome of the present work is that the linguistic phenomenon of colexification provides a valid basis for a knowledge-based computational text analysis method. A possible way to improve the measure presented in this work would be to use a more extensive colexification network as its basis, which would improve the coverage of words. Such an increased coverage would not only allow the model to capture a greater proportion of words from a text, but it would also make the algorithm more precise, since additional nodes and edges improve the calculation of the word similarity metric. One possible option for increasing the coverage would be to extend the Clics^3 database by adding new nodes. Those could either stem from other colexification databases or from automatically built sources, such as databases of identical translations. One very important aspect, however, is to ensure that the quality of the links in the network would not be significantly impacted by such modifications. Since the Clics^3 network was manually curated by experts from the field of linguistics, the creation of an automatically built database which has the same level of curation of Clics^3 might pose a great challenge.

A literature review in the field of NLP has shown that most text similarity measures integrating information from semantic networks such as WordNet or colexification networks incorporate statistical data from text corpora as well, thus requiring great amounts of data and qualifying as hybrid methods. In this work, we presented a method which is solely based on a semantic network. The genre recognition experiment has shown that our method performs with comparable accuracy as a standard NLP model and in some cases reaches significantly better predictions. Both models consider different characteristics of the texts they analyze, therefore these findings suggests that a combination of the two models might blend the strengths of both models, leading to a more complete and more efficient algorithm. One way to combine the two approaches consists in the addition of both text similarity estimates. More complex approaches could see the direct integration of term frequency data into the iterative greedy matching algorithm deployed in the colexification-based model. In general, we think that it would be of great interest to explore the hypothesis that both models complement each other, yielding an improved version of the colexification-based text analysis tool developed in this work.

Another interesting idea regarding future work consists in revisiting the author creativity experiment. As described in the previous chapters, we were not able to recreate the forward flow experiment reported in [GAC$^+$19] and thus validate the hypothesis that the colexification-based measure can be used to predict the creativity of a person from their association chains. Due to lack of data, which was not fully shared by the authors of the paper, we had to frame the experiment differently. In particular, we analyzed the data on a more granular level than what we initially intended. More in detail, instead of analysing the full association chains we considered only word pairs, which seems to have introduced noise. Indeed, it is easy to think that even a very creative person could answer to some of the seed words in the chain with a non creative answer. However, we expect that the mean similarity across the full association chain produced by a creative person would be higher than the one relative to a less creative person. We did not manage to prove this hypothesis because of unavailability of data. The noise introduced by the change in the experiment configuration of the task led to the final similarity prediction results not to differ significantly from a random predictor. We believe that this experiment could provide valuable insights if it were performed differently. In this case, the main aspect to change would definitely be the underlying data. A more extensive dataset of higher quality association chains (without misunderstandings nor repeated words in the same chain) would benefit the recreation of the experiment greatly. Optimally, future researchers should recreate the association chain experiment and the corresponding experiment to determine the creativity score for the participants. A good approach on how to test the creativity of participants can be found in [GAC$^+$19], that is by deploying psychologically accurate tests. We believe that that recreating this experiment using an extensive, reliable dataset could yield very interesting results about the relationship between creativity and semantic relationships between concepts and words. In particular, if the experiment would be successful this would imply that colexification patterns can be used to detect and quantify human psychological variables. On the opposite, a negative result would lead to the enquiry of whether word association tasks are related to the creativity of a person.

Future work based on the analysis presented in this thesis could also feature the application of the colexification-based method to other exploratory NLP tasks. One possible example of an additional task would be the analysis and classification of news articles reporting to historic events. One very interesting dataset for such an experiment is the 'DT Pilot Study Corpus'[1] of the Linguistic Data Consortium[2]. The dataset, originally created for 'topic detection and tracking' tasks includes various newswire texts from Reuters[3] and broadcast news from CNN[4], which were transcribed manually. The 16,000 texts of this corpus, balanced fairly between Reuters news texts and CNN broadcast transcripts, correspond to 25 different events from the period from July 1, 1994 to June 30, 1995. Furthermore, the dataset classifies events on the basis of their type. Based on this text corpus, it would be interesting to analyze how the topic of a text influences the performance of the colexification-based models in predicting how similar texts are. We assume that different text classes corresponding to different types of news events would accentuate characteristics peculiar to their class and therefore, different properties of the colexification-based measure. This would provide more insights into which topics tend to be more similar to each other and which type of concepts in the network are most valuable to identify semantic links between texts.

Another potential application of the colexification-based text analysis tool lies in the field of mental health detection and prevention. Indeed, we think that the identification of semantic relationships between words and texts could be a useful tool to explore the relationship between an authors style of writing and his mental health condition. Ideas about possible patterns and relationships between writing style and mental health can

[1] https://catalog.ldc.upenn.edu/LDC98T25
[2] https://www.ldc.upenn.edu/
[3] https://www.reuters.com/
[4] https://edition.cnn.com/

be found in psychology: various psycholinguistic studies claim that a writer's mood influences the range and type of topics addressed. Authors in positive moods tend to be more exploratory while authors suffering from mental health issues such as depression tend to focus on more narrow themes [Fre04]. This evidence might be connected to the idea of rumination, which is linked to depressive conditions. One idea about how to approach this question would be to analyze books from the Project Gutenberg corpus written by authors which committed suicide. It would be interesting to explore if such books cover a less broad range of topics and if parts of them are, on average, more similar to each other as opposed to other, random, parts of books. An interesting idea in this context is the concept of 'semantic breadth', i.e. the variety of topics addressed in a text. Lastly, another approach would be to analyze a set of tweets collected at the Complexity Science Hub[5]. Such a study would allow researchers to explore the extent to which colexification networks can be used in the analysis of very short texts. Furthermore, this might provide insights on which clusters of related concepts tend to be associated with the public conversation about mental health related topics on the internet.

[5] https://www.csh.ac.at/

List of Figures

List of Tables

Bibliography

[AD86] L Allison and T I Dix. A bit-string longest-common-subsequence algorithm. *Inf. Process. Lett.*, 23(6):305–310, 1986.

[ASCPCVS$^+$04] Tatiana Almeida Souza Coelho, Pável Pereira Calado, Lamarque Vieira Souza, Berthier Ribeiro-Neto, and Richard Muntz. Image retrieval using multiple evidence ranking. *IEEE Trans. on Knowl. and Data Eng.*, 16(4):408–417, 2004.

[BCCNEA20] Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. Tweeteval: Unified benchmark and comparative evaluation for tweet classification, 2020.

[BCjC19] Christine Basta, Marta R. Costa-jussà, and Noe Casas. Evaluating the underlying gender bias in contextualized word embeddings. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 33–39, Florence, Italy, 2019. Association for Computational Linguistics.

[BGMMS21] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery.

[BH01] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. *Workshop on WordNet and Other Lexical Resources*, 2001.

[BLL98] Curt Burgess, Kay Livesay, and Kevin Lund. Explorations in context space: Words, sentences, discourse. *Discourse Processes*, 25(2-3):211–257, 1998.

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, 2003.

[BTB14] Elia Bruni, Nam-Khanh Tran, and M. Baroni. Multimodal distributional semantics. *J. Artif. Intell. Res.*, 49:1–47, 2014.

[CCSB13] David Croft, Simon Coupland, Jethro Shell, and Stephen Brown. A fast and efficient semantic short text similarity metric. pages 221–227, 2013.

[CGHH91] Kenneth Church, William Gale, Patrick Hanks, and Don Hindle. *Using Statistics in Lexical Analysis*, pages 115–164. 1991.

[CM05] Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, Ann Arbor, Michigan, 2005. Association for Computational Linguistics.

[CRM99] N.A. Campbell, J.B. Reece, and L.G. Mitchell. *Biology*. Addison-Wesley world student series. Benjamin Cummings, 1999.

[CY05] Jung-Hsien Chiang and Hsu-Chun Yu. Literature extraction of protein functions using sentence pattern mining. *Knowledge and Data Engineering, IEEE Transactions on*, 17:1088– 1098, 2005.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. cite arxiv:1810.04805Comment: 13 pages.

[DDF$^+$90] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R.A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, pages 391–407, 1990.

[DNPG21] Anna Di Natale, Max Pellert, and David Garcia. Colexification networks encode affective meaning. *Affective Science*, 2021.

[ER04] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, 2004.

[Eri17] Lars Eriksen. Die polysemie in der allgemeinsprache und in der juristischen fachsprache. oder: Zur terminologie der "sache" im deutschen. *HERMES - Journal of Language and Communication in Business*, 15:211, 2017.

[FK79] W. N. Francis and H. Kucera. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979.

[FKL98] Peter W Foltz, Walter Kintsch, and Thomas K Landauer. The measurement of textual coherence with latent semantic analysis. *Discourse processes*, 25(2-3):285–307, 1998.

[Fra08] Alexandre François. Semantic maps and the typology of colexification: Intertwining polysemous networks across languages. In Martine Vanhove, editor, *From Polysemy to Semantic change: Towards a Typology of Lexical Semantic Associations*, Studies in Language Companion Series, pages 163–215. Benjamins, 2008.

[Fre04] Barbara L. Fredrickson. The broaden-and-build theory of positive emotions. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 359(1449):1367–1378, 2004. 15347528[pmid].

[FV15] Ingrid Lossius Falkum and Agustin Vicente. Polysemy: Current perspectives and approaches. *Lingua*, pages 10–1016, 2015.

[GAC+19] Kurt Gray, Stephen Anderson, Eric Chen, John Kelly, Michael Christian, John Patrick, Laura Huang, Yoed Kenett, and Kevin Lewis. "forward flow": A new measure to quantify free thought and predict creativity. *American Psychologist*, 74:539–554, 2019.

[GF13] Wael Gomaa and Aly Fahmy. A survey of text similarity approaches. *international journal of Computer Applications*, 68, 2013.

[GFC20] Martin Gerlach and Francesc Font-Clos. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1), 2020.

[GvdHAK+14] Joaquín Goñi, Martijn P. van den Heuvel, Andrea Avena-Koenigsberger, Nieves Velez de Mendizabal, Richard F. Betzel, Alessandra Griffa, Patric Hagmann, Bernat Corominas-Murtra, Jean-Philippe Thiran, and Olaf Sporns. Resting-brain functional connectivity predicted by analytic measures of network communication. *Proceedings of the National Academy of Sciences*, 111(2):833–838, 2014.

[GVH+16] Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas, 2016. Association for Computational Linguistics.

[Haa00] Kenneth B. Haase. Interlingual BRICO. *IBM Syst. J.*, 39(3&4):589–596, 2000.

[HG14] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014.

[HKE99] Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[HRK15] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 2015.

[HSO98] Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press, 1998.

[HWSU20] Sam Henry, Yanshan Wang, Feichen Shen, and Ozlem Uzuner. The 2019 National Natural language processing (NLP) Clinical Challenges (n2c2)/Open Health NLP (OHNLP) shared task on clinical concept normalization for clinical records. *Journal of the American Medical Informatics Association*, 27(10):1529–1537, 2020.

[II08] Aminul Islam and Diana Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2):1–25, 2008.

[JC97] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference*, pages 19–33, Taipei, Taiwan, 1997. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

[JM09] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA, 2009.

[JWH$^+$19] Joshua Conrad Jackson, Joseph Watts, Teague R. Henry, Johann-Mattis List, Robert Forkel, Peter J. Mucha, Simon J. Greenhill, Russell D. Gray, and Kristen A. Lindquist. Emotion semantics show both cultural variation and universal structure. *Science*, 366(6472):1517–1522, 2019.

[Kle02] Ekaterini Klepousniotou. The processing of lexical ambiguity: Homonymy and polysemy in the mental lexicon. *Brain and Language*, 81(1):205–223, 2002.

[Kow97] Gerald Kowalski. *Information Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers, USA, 1st edition, 1997.

[KPS04] Youngjoong Ko, Jinwoo Park, and Jungyun Seo. Improving text categorization using the importance of sentences. *Information Processing Management*, 40:65–79, 2004.

[Kro97] Robert Krovetz. Homonymy and polysemy in information retrieval. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 72–79, Madrid, Spain, 1997. Association for Computational Linguistics.

[LB02] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[LB05] Mirella Lapata and Regina Barzilay. Automatic evaluation of text coherence: Models and representations. pages 1085–1090, 2005.

[LBHS07] Wolfgang Lenhard, Herbert Baier, Joachim Hoffmann, and Wolfgang Schneider. Automatische bewertung offener antworten mittels latenter semantischer analyse. *Diagnostica*, 53:155–165, 2007.

[LBM03] Yuhua Li, Zuhair A. Bandar, and David McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):871–882, 2003.

[LC98] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[Les86] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, page 24–26, New York, NY, USA, 1986. Association for Computing Machinery.

[LG05] Tao Liu and Jun Guo. Text similarity computing based on standard deviation. In De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Advances in Intelligent Computing*, pages 456–464, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[LGA$^+$18] Johann-Mattis List, Simon Greenhill, Cormac Anderson^1, Thomas Mayer^3, Tiago Tresoldi, and Robert Forkel^1. Clics^2 an improved database of cross-linguistic colexifications : Assembling lexical data with the help of cross-linguistic data formats. *Linguistic Typology*, 22, 2018.

[Lin02] Dekang Lin. Extracting collocations from text corpora. 2002.

[LLD$^+$20] Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online, 2020. Association for Computational Linguistics.

[LMB$^+$06] Yuhua Li, David McLean, Zuhair Bandar, James O'Shea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18:1138–1150, 2006.

[LSM13] Minh-Thang Luong, Richard Socher, and Christopher D. Manning. Better word representations with recursive neural networks for morphology. Sofia, Bulgaria, 2013.

[LSMHL19] Philipp Lorenz-Spreen, Bjarke Mønsted, Philipp Hövel, and Sune Lehmann. Accelerating dynamics of collective attention. *Nature Communications*, 10, 2019.

[LZ04] Ying Liu and Chengqing Zong. Example-based chinese-english mt. volume 7, pages 6093 – 6096 vol.7, 2004.

[MBF$^+$90] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to wordnet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244, 1990.

[MBK00] C.T. Meadow, B.R. Boyce, and D.H. Kraft. *Text Information Retrieval Systems*. Library and information science. Academic Press, 2000.

[MC91] George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

[MCS06] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, page 775–780. AAAI Press, 2006.

[MR86] James L. McClelland and D. E. Rumelhart. Mechanisms of sentence processing: Assigning roles to constituents of sentences. 1986.

[NK19] Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy, 2019. Association for Computational Linguistics.

[NMS11] Douglas L Nelson, Cathy L McEvoy, and Th A Schreiber. University of south florida free association norms. *URL: http://w3. usf. edu/FreeAssociation/(: 08.04. 2017)*, 2011.

[OBCM08] James O'Shea, Zuhair Bandar, Keeley Crockett, and David McLean. A comparative study of two short text semantic similarity measures. volume 4953, pages 172–181, 2008.

[OMMI03] Naoaki Okazaki, Yutaka Matsuo, Naohiro Matsumura, and Mitsuru Ishizuka. Sentence extraction by spreading activation with refined similarity measure. pages 407–411, 2003.

[Pan82] Otto Panman. Homonymy and polysemy. *Lingua*, 58(1):105–136, 1982.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999. Previous number = SIDL-WP-1999-0120.

[PRJ05] Eui-Kyu Park, Dong-Yul Ra, and Myung-Gil Jang. Techniques for improving web retrieval effectiveness. *Information Processing Management*, 41:1207–1223, 2005.

[Res95] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, page 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[Res99] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Int. Res.*, 11(1):95–130, 1999.

[RH21] Navid Rekabsaz and Allan Hanbury. An unbiased approach to quantification of gender inclination using interpretable word representations. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2021.

[Rin18] Tyler W. Rinker. *textstem: Tools for stemming and lemmatizing text*. Buffalo, New York, 2018. version 0.1.4.

[Roc71] J. J. Rocchio. *Relevance Feedback in Information Retrieval*. Prentice Hall, Englewood, Cliffs, New Jersey, 1971.

[RPPV20] Max Ryabinin, Sergei Popov, Liudmila Prokhorenkova, and Elena Voita. Embedding words in non-vector space with unsupervised graph learning, 2020.

[RTG+20] Christoph Rzymski, Tiago Tresoldi, Simon Greenhill, Mei-Shin Wu, Nathanael Schweikhard, Maria Koptjevskaja-Tamm, Volker Gast, Timotheus Bodt, Abbie Hantgan, Gereon Kaiping, Sophie Chang, Yunfan Lai, Natalia Morozova, Heini Arjava, Nataliia Hübler, Ezequiel Koile, Steve Pepper, Mariann Proos, Briana Epps, and Johann-Mattis List. The database of cross-linguistic colexifications, reproducible analysis of cross-linguistic polysemies. *Scientific Data*, 7, 2020.

[Sch98] Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

[SGM19] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy, 2019. Association for Computational Linguistics.

[SL68] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *J. ACM*, 15(1):8–36, 1968.

[SM86] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., USA, 1986.

[Sow91] John Sowa, editor. *Principles of Semantic Networks: Explorations in the Representation of Knowledge (Morgan Kaufmann Series in Representation and Reasoning)*. Morgan Kaufmann Pub, May 1991.

[SSMB97] Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. Automatic text structuring and summarization. *Information Processing Management*, 33(2):193–207, 1997. Methods and Tools for the Automatic Construction of Hypertext.

[SV21] Robert Stewart and Sumithra Velupillai. Applied natural language processing in mental health big data. *Neuropsychopharmacology*, 46(1):252–253, 2021.

[Tar09] Sven Tarp. Homonymy and polysemy in a lexicographical perspective. *Zeitschrift für Anglistik und Amerikanistik*, 57, 2009.

[Tur50] A. M. Turing. I.—Computing Machinery And Intelligence. *Mind*, LIX(236):433–460, 1950.

[Tur01] Peter D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In Luc De Raedt and Peter Flach, editors, *Machine Learning: ECML 2001*, pages 491–502, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[WBS09] Frank Walter, Stefano Battiston, and Frank Schweitzer. Personalised and dynamic trust in social networks. *RecSys'09 - Proceedings of the 3rd ACM Conference on Recommender Systems*, 2009.

[WMSS12] P. Waila, Marisha, V. Singh, and M. Singh. Evaluating machine learning and unsupervised semantic orientation approaches for sentiment analysis of textual reviews. *2012 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–6, 2012.

[WP94] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, ACL '94, page 133–138, USA, 1994. Association for Computational Linguistics.

[WPN^{+}19] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[WW64] Uriel Weinreich and Webster. Webster's third: A critique of its semantics. *International Journal of American Linguistics*, 30(4):405–409, 1964.

[XRK^{+}21] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Theoretical understandings of product embedding for e-commerce machine learning. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 256–264, New York, NY, USA, 2021. Association for Computing Machinery.

[YSS^{+}15] Hyejin Youn, Logan Sutton, Eric Smith, Cristopher Moore, Jon Wilkins, Ian Maddieson, William Croft, and Tanmoy Bhattacharya. On the universal structure of human lexical semantics. *Proceedings of the National Academy of Sciences*, 113, 2015.

[YTPM11] Wen-Tau Yih, Kristina Toutanova, John Platt, and Christopher Meek. Learning discriminative projections for text similarity measures. pages 247–256, 2011.