EBERHARD-KARLS-UNIVERSITÄT TÜBINGEN

MASTER'S THESIS

A graphical Approach to unsupervised Dictionary Induction

Author:
Tobias Konrad ELSSNER
Student Number: 3751602

Supervisors:
Prof. Dr. Kurt EBERLE
Dr. Çağrı ÇÖLTEKIN

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Arts

Philosophische Fakultät
Seminar für Sprachwissenschaft

6.8.2020

Hiermit versichere ich, dass ich die Arbeit selbständig verfasst, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt, alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe und dass die Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist und dass die Arbeit weder vollständig noch in wesentlichen Teilen bereits veröffentlicht wurde sowie dass das in Dateiform eingereichte Exemplar mit den eingereichten gebundenen Exemplaren übereinstimmt.

I hereby declare that this paper is the result of my own independent scholarly work. I have acknowledged all the other authors' ideas and referenced direct quotations from their work (in the form of books, articles, essays, dissertations, and on the internet). No material other than that listed has been used.

Tübingen, August 6, 2020

Firstname Surname

Contents

List of Figures

List of Tables

List of Abbreviations

MT	**M**achine **T**ranslation
NLP	**N**atural **L**anguage **P**rocessing
SVD	**S**ingular **V**alue **D**ecomposition
LSA	**L**atent **S**emantic **A**nalysis
RI	**R**andom **I**ndexing
NN	**N**eural **N**etwork
RNN	**R**ecursive **N**eural **N**etwork
CBOW	**C**ontinuous **B**ag-**O**f-**W**ords
SGD	**S**tochastic **G**radient **D**escent
BP	**B**ack **P**ropagation
NEG	**NEG**ative Sampling
OOV	**O**ut **O**f **V**ocabulary
FSA	**F**inite **S**tate **A**utomata
FST	**F**inite **S**tate **T**ransducer
CCA	**C**anonical **C**orrelation **A**nalysis
GAN	**G**enerative **A**dversarial **N**etwork
CSLS	**C**ross **D**omain **S**imilarity **L**ocal **S**caling
ISF	**I**nverted **S**o**f**t-max
PPR	**P**ersonalized **P**age**R**ank

Chapter 1

Motivation

"I only speak two languages: English and bad English."
Korben Dallas, The Fifth Element (Besson and Kamen, 1997)

Machine translation is one of the most challenging fields in computational linguistics. Not only does it involve understanding semantics, morphology, and syntax in one, but also transferring this information to another language. The more data knowledge is acquired for one par of languages, the easier becomes the translation process.

Conservatively estimated, there exist over 6800 distinct languages (Anderson, 2010). The `World Atlas of Language Structures` (Dryer and Haspelmath, 2013) currently lists 2,662[1] languages world wide, which are extensively studied, categorized, and publicized.

At the same time, GOOGLETRANSLATE offers its services for a fraction of 108 of those 2662 - that is, about 4% - of these languages[2]. DEEPL only supports eleven languages[3]. While the Internet has enabled communication around the world, it is still difficult to communicate without speaking an interlocutor's language or a common *lingua franca*, for instance English, fluently.

The motivation for this thesis is to facilitate an essential part of bilingual communication, particularly the construction of dictionaries. Although being linguistically studied, many languages come with low resources in terms of human experts or (written) data. However, compiling a dictionary usually requires both: Translators for the featured languages, as well as data to extract distinct use-cases of words.

Therefore, this project conducts and evaluates a methodology to induce a dictionary, which is especially designed for a small data set, without *any* foreknowledge on translations between the languages. So, the questions arise, how words need to be represented, what a mapping in between could look like, and how it can be calculated. The subsequent chapters address these considerations.

Chapter 2 gives a brief overview on general approaches to automated translation. In Chapter 3, the meaning of words and their representation is investigated. Chapter 4 shows how dictionaries can be induced, and presents the method advocated in this .

[1] `https://wals.info/languoid` [Accessed: 5.8.2020]
[2] `https://translate.google.com/` [Accessed: 5.8.2020]
[3] `https://www.deepl.com/translator` [Accessed: 5.8.2020]

In Chapter 5, the method is evaluated and compared to related work, and Chapter 6, discusses the results and gives ideas for future work.

Chapter 2

Approaches to Machine Translation

"'The Babel fish,' said The Hitchhiker's Guide to the Galaxy quietly, 'is small, yellow and leech-like, and probably the oddest thing in the Universe.'"
The Hitchhiker's Guide to the Galaxy, Chapter 6 (Adams, 1979)

Starting with the Weaver-Memorandum in 1947 (Weaver, 1955), the task of machine translation (henceforth, MT) gained widespread attention (Arnold et al. (1994), chapter 1.4). Since then, three important streams have emerged, depending on the perspective on language. That is, in chronological order, either a system of formal rules, by which sentences are constructed and translated (cf. Arnold et al. (1994) section 10.2 for an introduction, and Galley et al. (2004)) or phrases shallowly recombined and transduced (see example-based MT frameworks, such as Franz et al. (2000) or Gough and Way (2004)), a (generative) statistical process (read chapter 10.4.2 as an outline by Arnold et al. (1994), Brown et al. (1990), Och and Ney (2002), and Och and Ney (2004)), a connectivist-driven dense continuous representation (Koehn (2017) gives a general overview, Bahdanau et al. (2014), Cho et al. (2014)), or hybrid approaches (for instance, Wu (1997), Ayan et al. (2004), Thurmair (2005), and Zou et al. (2013)). Regardless of the chosen methodology, three main difficulties of automated translation need to be taken into account, as Arnold et al. (1994) point out in chapter 6:

Lexical and Structural Ambiguity
Words with more than one meaning are said to be lexically ambiguous, phrases with more than one reading are called structural ambiguous. Both ambiguities pose problems to MT systems; the number of possible translations becomes multiplied. Solutions could include contextual disambiguation, as Weaver (1955) suggests, or world knowledge from ontologies, (such as FrameNet (Baker et al., 1998), GermaNet (Hamp and Feldweg, 1997) or WordNet (Miller, 1995)). For instance, the English word *use* can be a verb, as well as a noun, and can be appropriated in multiple unrelated contexts.

Lexical & Structural Mismatches
Languages differ in the way how the space of meaning is partitioned by words and grammatical functions. A word in one language can have multiple translations in another, depending on its context; these translations can either consist

of one term, or is represented by a phrase. An example would be the French verb *ignorer*, whose English translation *to not know* or *to be ignorant of* need to be expressed by a phrase. Furthermore, the same syntactic structures in two languages do not have to correspond: The passive sentence *He is called Sam* translates to the indicative *Er heißt Sam*, and the auxiliary construction *He likes to swim* becomes the adverbial phrase *Er schwimmt gerne*.

Idioms & Collocations

Idioms are expressions whose meaning cannot be composed by their subparts alone. Once the idiomatic meaning is recovered, the problem is to find a corresponding idiom in the target language, or, if such does not exist, determine an appropriate paraphrase. For instance, the figure of speech *to kick the bucket* translates to *casser sa pipe*, or *mourir*, in French.

Collocations are two or more words, whose meaning is, in contrast to idioms, compositional, but which can hardly be replaced by similar or synonymous. A *heavy smoker* becomes in German a *starker Raucher*, as opposed to **schwerer Raucher*.

These considerations have to be taken into account when designing a applicable MT program.

What is quietly presumed, is a comprehensive dictionary with wide coverage of topics. In rule-based systems, this task is manually accomplished by human experts. Niehues and Waibel (2012) gives an overview on how statistical methods can induce such dictionaries. In neural network approaches, either joint embeddings (as described by Zou et al. (2013)), or functional mappings from the source to target language can be viewed as phrase-table.

In all frameworks presented so far, human knowledge in terms of prewritten rules or corpora is indispensable. The most common type of resources are parallel corpora, meaning identical texts in multiple languages, where the sentences are being aligned. However, compiling rules, ontologies and parallel corpora requires on the one hand skilled translators, on the other hand vast data collections, and both aspects are costly and time consuming.

This thesis here aims to mitigate the dependence on predefined knowledge. It follows the current trend of dense vector representations, which are aligned in unsupervised fashion to form a dictionary, which can be employed by the aforementioned methods. The most prevalent problem will be therefore *lexical ambiguities*.

The next chapter presents why, and how, word vectors are obtained.

Chapter 3

Word Meaning and Representation

"You shall know a word by the company it keeps."
John Rupert Firth (Firth, 1957)

When constructing an interlingual dictionary in practice, the first task is how the meaning of entries can be determined and represented. This chapter focuses only on practical aspects; formal semantic or pragmatic and cognitive aspects of word meaning are omitted at this point. Interested readers are referred to the books by Zimmermann and Sternefeld (2013) and Noveck and Sperber (2004).
Closely connected to the question how lexical units are represented, is how they are interrelated. After all, similar words should to be translated alike. As the goal of this thesis is to explore a novel unsupervised translational approach especially meant for small data sets, syntactic information, for example as objectival or prepositional relations, is initially not integrated. Reasons for this exclusion are twofold: First, the avoidance of an potential error source, which possibly complicates the evaluation at an early stage as second, small data sets are often not guaranteed to contain full variance.
Practically worth considering are two methodologies: Logical ontologies, where each word is manually assigned to a fixed number of (semantic) roles and meanings, and distributional semantic methods, where the word meaning is defined by its context. No matter which approach is chosen, it has to yield a distance function between terms, because otherwise relations between words could not be established. Based on such relationships, interlingual similarities are later exploited, in order to find corresponding terms in different languages.
The following chapter presents these approaches, with their advantages, downsides, and how they compare to each other. Some sections are consciously written more verbose, as certain concepts are later re-used in the induction step.

3.1 Linguistic Ontologies

Originally meant as a resource of world knowledge for artificial intelligence systems, linguistic databases such as `WordNet` (Miller et al., 1990), its derivatives `EuroWordNet` (Vossen, 2002) and particularly `GermaNet` (Hamp and Feldweg, 1997), as well as

FrameNet (Baker et al., 1998) provide detailed semantic information about ontological relations between lexical entities. Besides abstract relationships, concrete use-cases are given by 'naturalistic corpora' in FrameNet (Ruppenhofer et al. (2006), page 6), glosses in WordNet and its descendents (see Miller (1995) and Kunze and Lemnitzer (2002)) and sense tags computed on their basis (Harabagiu et al. (1999) and Henrich et al. (2011)).
Since these databases are continuously developing for more than two decades, it is impossible to describe them to the full extend. That is why, this subsection can only list a subset of their main features, together with their advantages and downsides. For a detailed review, readers are redirected to the original publications.
Sharing a similar blueprint, WordNet and GermaNet are built on the same kind of hierarchical categorization and relationships between so-called syn-sets(Miller (1995), Hamp and Feldweg (1997), and Vossen (2002)). Each entry is either noun, verb, adjective or adverb. Each of those part-of-speech tags is then again sub-categorized into fifteen semantic fields. The basic relations are summarized below (see Miller et al. (1990) pages 45-55, and Hamp and Feldweg (1997)).

Relation	Description	*Example*
Synonymy	Two {noun, verb, adjective, adverb}s that can be used interchangeably in the same context.	*pipe* $\leftrightarrow$ *tube*
Antonymy	Inverse **Synonymy** relation.	*wet* $\leftrightarrow$ *dry*
Hypernomy	One noun that is super-ordinated to another one.	*tree* $\rightarrow$ *maple*
Hyponymy	Inverse **Hypernymy** relation.	*maple* $\rightarrow$ *tree*
Holonomy	One noun includes another one as a part.	*fleet* $\rightarrow$ *ship*
Meronymy	Inverse **Holonomy** relation.	*ship* $\leftarrow$ *fleet*
Troponymy	One verb describes another one in a certain manner.	*march* $\rightarrow$ *walk*
Entailment	One verb is entailed by the other's action.	*drive* $\leftrightarrow$ *ride*
Cause	One Verb (the *causative*) causes another verb (the *resultative*) to be.	*teach* $\rightarrow$ *learn*

TABLE 3.1: Overview over the basic Relation types in WordNet and GermaNet

Based on this list of elementary relations, Vossen (2002) describe some more detailed, as **Co_Role**, **Has_Subevent** or **Is_Subevent_Of** (see Vossen (2002), section 2.2). Additionally, GermaNet has annotated selectional restrictions, which give "information about typical nominal arguments for verbs and adjectives" (Hamp and Feldweg, 1997). If not present, sub- or superordinated groups are created artificially, such as *?educated human* in (Hamp and Feldweg, 1997). For more detailed information on the structural design, readers are referred to Miller et al. (1990), Vossen (2002) and Kunze and Lemnitzer (2002).
FrameNet differs from the other *-Net databases by also considering prepositions (Ruppenhofer et al. (2006), page 45). Annotations consist of three parts: A "frame element (for example, Food), a grammatical function(say, Object) and a phrase type(say, NP)." (Ruppenhofer et al. (2006), page 6). While frame elements denotes a domain, and phrase types their PoS-tag, grammatical functions "describe the ways in which the constituents satisfy abstract grammatical requirements of the target word" as

Ruppenhofer et al. (2006) note on page 63. The complete list is given in chapter 5.1 in (Ruppenhofer et al., 2006), including **object**, **dependent**, **external argument**, and **modified head noun**.

The ⟨frame element, function, phrase type⟩ triples then themselves form the frame of each annotated sentence (Ruppenhofer et al. (2006), page 6). Frames are interconnected by so-called frame relations:

Relation	Description	*Example*
Inheritance	Two frames share a Is-a relationship.	*car* → *vehicle*
Perspective_on	Two frames that can be looked at from the same perspective.	*Get_a_job* ↔ *Hiring*
Subframe	Subordinated frames that belong to complex structured frames.	*Arrest*, *Trial* → *Criminal_process*
Precedes	One subframe {logically, temporarily} precedes another.	*Being_awake* → *Fall_asleep*
Inchoative_of	One frame triggers another one.	*to rise* → *Change_position_on_a_scale*
Causative_of	One frame causes another one.	*to raise* → *Cause_change_of_scalar_position*
Using	One child frame invokes the parent frame.	*Volubility* → *Communication*
See_also	Similar frames with subtle semantic differences.	*Scrutiny* ↔ *Seeking*

TABLE 3.2: Overview over the Relation types in `FrameNet` ((Ruppenhofer et al., 2006), chapter 6.1)

As can be seen from Table 3.2, relations in `FrameNet` show a higher degree of abstraction than the ones in `WordNet`. On the downside, simple, though important relationships like **synonymy** are not readily available. `FrameNet`'s developers try to bypass this downside by incorporating missing relations from `WordNet` (Ruppenhofer et al. (2006), page 86).

Ontologies are a promising rich database about world knowledge. The level abstraction can hardly by reconstructed by automatic systems. Thanks to human evaluation, especially polysemy is well handled.

But while ontologies exhibit insightful information on the semantic side, it is for several reasons difficult to exploit for dictionary induction. Firstly, the annotation for both languages has to be consistent. A comparison of, for instance, `FrameNet` and `GermaNet` would be technically infeasible. Secondly, even if the same annotation is available for multiple languages, as in the case of `EuroWordNet`, the problem of defining a distance function persists. Even though it is possible to take the number of steps from syn-set to syn-set between two entries as a distance measure, this method would be coarse-grained: Due to the binary nature of relationships, by which two syn-sets are either related or not, any non-existence of a relation for an entry in one language leads to a significantly lower similarity score, even if the two bilingual entries are semantically corresponding. Thirdly, more artificially, the goal is to induce a dictionary in an unsupervised fashion. Particularly, for languages with fewer resources, building ontologies in first place is very tedious. Creating an unsupervised method that relies on such elaborate databases would not be reasonable. Thus, the next section presents approaches for an automated representation of word meaning; less abstractive on the one hand, but more practical on the other.

3.2 Distributional Semantics

The idea behind distributional semantics is that the meaning of a word comprises of its context. As Wittgenstein already notes 1953 in §43 of (Wittgenstein, 1953), " the meaning of a word is its use in the language." Harris (1954) formalizes this view in chapter *Meaning as a function of distribution*, stating "[I]f A and B have almost identical environments except chiefly for sentences which contain both, we say they are synonyms: *oculist* and *eye-doctor*. If A and B have some environments in common and some not (e.g. *oculist* and *lawyer*) we say that they have different meanings, the amount of meaning difference corresponding roughly to the amount of difference in their environments." The importance of context is also integrated into ontologies, by providing the user with example sentences for each entry. The upcoming sections now discuss various approaches based on the distributional assumption. All have in common that a word is represented by a vector, whose entries define some kind of meaning - either a concrete word it co-occurs with, or some abstract concept. That is, each word becomes a data-point in a high-dimensional vector space.
Before diving into the details of word quantization methods, it is worth noting the change in the characterization of word meaning. In contrast to linguistic ontologies, which *explicitly* incorporate multiple relationships between words, similarity between word vectors is now a function mapping from $\mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$. On the one hand, this simplifies the process of dictionary induction later on, as all possible relations are *implicitly* bundled in one number. On the other hand, certain relationships, for instance troponymy or entailment, which might enhance the accuracy of translations, are hard to recover from a single digit. To emphasize this shift, the most common measures are presented, with an adapted notation from Bullinaria and Levy (2007). In all cases, similarity is calculated between two words w_1, w_2, and their associated vectors $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^m$.
As a starting point, evident similarity measures between are standard distance computations between vectors, such as Manhattan or Euclidean distance. Also called city-block-metric,

$$m(w_1, w_2) = \sum_{i=1}^{m} |\mathbf{w}_1[i] - \mathbf{w}_2[i]| \tag{3.1}$$

the Manhattan distance returns the sum of absolute distances between the entries, in analogy to the distance one would have to walk from one point to another in a city subdivided into rectangular block, while the Euclidean distance

$$e(w_1, w_2) = \sqrt{\sum_{i=1}^{m} (\mathbf{w}_1[i] - \mathbf{w}_2[i])^2} \tag{3.2}$$

measures the distance in terms of 'shortcuts'. $d(\cdot, \cdot)$ is zero, if w_1 and w_2 are identical, and grows as both vectors diverge. The main drawback is its dependence on vector length: Two semantically similar word vectors can result in a low score, if they appear in identical context, but one occurs much more often than the other.

The cosine similarity remedies this downside:

$$cos(w_1, w_2) = \frac{\langle \mathbf{w}_1, \mathbf{w}_2 \rangle}{|\mathbf{w}_1\|_2 \, |\mathbf{w}_2\|_2} = \frac{\sum\limits_{i=1}^{m} \mathbf{w}_1[i] \; \mathbf{w}_2[i]}{\sqrt{\sum\limits_{i=1}^{m} \mathbf{w}_1[i]^2} \sqrt{\sum\limits_{i=1}^{m} \mathbf{w}_2[i]^2}} \tag{3.3}$$

The nominator normalizes the dot-product of both vectors $\langle \cdot, \cdot \rangle$ in the numerator by the product of their lengths. Thus, the outcome is length-independent and bound between -1 and 1. The measurement can be visualized as the cosine of the angle between w_1 and w_2; -1 corresponds to an angle of 180 degrees, meaning the vectors point into opposite directions, 0 to 90 degrees, i. e. the vectors are orthogonal to each other, and 1 refers to an angle of zero degrees, meaning the vectors are identical.
Another possibility is to use probabilities instead of raw entries. Then, the word vectors are normalized with regards to the sum of their values:

$$\mathbf{w}[i] = \frac{\mathbf{w}[i]}{\sum\limits_{i'=1}^{m} \mathbf{w}[i']} \tag{3.4}$$

If some entries are negative, the soft-max function can be employed:

$$\mathbf{w}[i] = \frac{e^{\mathbf{w}[i]}}{e^{\sum\limits_{i'=1}^{m} \mathbf{w}[i']}} \tag{3.5}$$

Doing so mitigates the effect of the exceptionally high raw counts, as all entries of a word vector sum up to one. Forming a probability distribution for each vector, the normalization also allows the usage of probabilistic distance functions, such as Kullback-Leibler divergence, Hellinger or Bhattacharya distance. Representatively for the aforementioned, the Kullback-Leibler is presented, as it is one of the most widely used, not only in word similarity tasks.

$$kl(w_1, w_2) = \sum_{i=1}^{m} \mathbf{w}_1[i] \cdot \log_2 \frac{\mathbf{w}_1[i]}{\mathbf{w}_2[i]} \tag{3.6}$$

The divergence calculates the *expected* amount of bits being additionally necessary when w_1 is encoded by the distribution over the context of w_2. By the rules for logarithmic calculation, it follows that $\mathbf{w}_1[i], \mathbf{w}_2[i] \overset{!}{>} 0, \; \forall \, i$. Interestingly, the KL-divergence is generally non-symmetric, meaning that $kl(w_1, w_2) \neq kl(w_2, w_1)$. What seems first counterintuitive, can be a neat feature, for instance for modeling word association; *lawn* might be more associated with *green*, than vice versa.
One type of similarity which has not been discussed yet is *attributional* similarity. Hereby, the similarity *between* similarities is measured. Let w_1, w_2, w_3, w_4 be terms, and let $w_1 : w_2$ and $w_3 : w_4$ denote a relation between term one (three) and term two (four). Then, for some similarity measure $sim(\cdot, \cdot)$, Turney (2006) propose a meta

score for the two similarities as follows:

$$score(w_1 : w_2 :: w_3 : w_4) = \frac{1}{2}\left(sim(\mathbf{w}_1, \mathbf{w}_2) + sim(\mathbf{w}_3, \mathbf{w}_4)\right) \tag{3.7}$$

If w_4 is not given, as for instance in SAT-Tests, it can be calculated by

$$w_4 = \arg\max_{w_?}\ score(w_1 : w_2 :: w_3 : w_?) \tag{3.8}$$

Mikolov et al. (2013b) take a similar approach, by defining missing $w_?$ as

$$\mathbf{w}_? = \mathbf{w}_2 - \mathbf{w}_1 + \mathbf{w}_3 \tag{3.9}$$

Since $\mathbf{w}_?$ might not match a word vector in the vocabulary, Mikolov et al. (2013b) use the closest vector according to cosine similarity:

$$w_4 = \arg\max_{w_?}\ cos(w_2 - w_1 + w_3, w_?). \tag{3.10}$$

The key note here is that in both cases attributional relations are thought to be linear. That view is supported by the results of Mikolov et al. (2013) and Mikolov et al. (2013a), stating that "non-linear models also have a preference for a linear structure of the word representations"(Mikolov et al., 2013a). Later on, this inherent linear structure becomes important, when subword information is included (see Section 3.2.2.3).
Although lacking linguistic insight into the kind of relations, the similarity functions presented here offer a more general perspective on how words can be organized in terms of meaning. In the remainder of this chapter, various approaches to the construction of word vectors are discussed. Results are thereby included, to make transparent which method is preferred to others.

3.2.1 Counting-Based Models

Being among the first practically explored distributional models (Rubenstein and Goodenough, 1965), counting-based methods describe the meaning of a word solely based on how often it co-occurs with other terms in natural language texts within a certain window. Hence, a straightforward representation for a set of n words is a co-occurrence matrix over reals, $\mathbf{M} \in \mathbb{R}^{n\times n}$, where an arbitrary entry $\mathbf{M}[i][j]$ stands for number of times the ith word w_i co-occurs with jth term w_j. Obviously, $\mathbf{M}$ is symmetric, meaning $\mathbf{M}[i][j] = \mathbf{M}[j][i]$. The word vector for term w_i, $\mathbf{M}[i][:]$, is henceforth more generally indicated by $\mathbf{w}_i$. The size of the window can be chosen freely; however, it is useful to employ a distance metric to decrease the importance of distant terms.
Keeping in mind that each word equates to one dimension, count-based models are prone to result in sparse matrices, meaning that many entries will be (close to) zero. That is, because most words usually have a limited number of contexts to appear

in (Harris, 1954). If the set of context words of two otherwise similar terms incidentally differs by one word, this can result in undesired changes in the similarity score. Conversely, similarity measures can be mislead by (functional) words occurring in the contexts of many words; for instance, the determiner *the* appears in the context of many nouns, which do not share a similar meaning. There are two ways how these problems are tackled: On the one hand, by a careful design of weighting functions, and on the other hand, by selecting only important dimensions. The upcoming subsections give an overview about the methods.

3.2.1.1 Weighting Functions

The necessity of weighting functions arises from the observation that word similarities are skewed by irrelevant contexts, as determiners or prepositions. Two otherwise alike terms might only differ in their grammatical gender (cf. German:*der* Schrank versus *das* Regal), which results in a lower similarity score. In order to tackle this problem, various weighting functions have been proposed. Their commonality is that they measure how surprising a certain unit appears in a words' context; the more unexpected a context is, the more relevant it is for the word in question. This brief overview presents exemplarily two widely-used functions, $tf.idf$ and *pmi*, based on Bullinaria and Levy (2007) and Turney and Pantel (2010).
$tf.idf$, short for term-frequency multiplied by inverted (logarithmic) document frequency, was first investigated in information retrieval research by Sparck Jones (1972) and defines a whole family of weighting functions (Salton and Buckley, 1988). Applying the notation of Salton and Buckley (1988) to term-term matrices, the $tf.idf$-weight is calculated by

$$tf.idf(w_i, w_j) = tf(w_i) \cdot \log\left(\frac{n}{\sum\limits_{\mathbf{M}[i'][j] \neq 0} \mathbf{1}}\right) \tag{3.11}$$

$tf(w_i)$ can either denote raw or normalized (either by sum over all its entries, or its vector length) term occurrences, w_j a particular context word, and n the number of all terms in the model. The ratio inside the logarithm calculates the inverse of the probability that context w_j appears with another word $w_{i'}$ by chance.
Taking the logarithm of the ratio originates in Shannon's information-theoretic interpretation of *entropy* (cf. chapter six 'Choice, Uncertainty and Entropy' in (Shannon, 1948)). There, it calculates the number of bits necessary to encode a event i in a

discrete distribution X with probability p_i :

$$\begin{aligned} H(X) &= -\sum_{i \in X} p_i \log(p_i) \\ &= \sum_{i \in X} p_i \log(p_i^{-1}) \\ &= \sum_{i \in X} p_i \log(\frac{1}{p_i}) \end{aligned} \tag{3.12}$$

$H(X)$ computes the *expected* number of bits needed in the encoding of X. It follows from logarithmic laws that only events with non-zero probabilities are accounted for. So, from a theoretical perspective, *tf.idf* weights the frequency/ probability of a term by the length of the bit sequence encoding of a certain context word. From a practical standpoint, this means that the weight becomes higher, the rarer the context word appears together with other terms. It even can be zero, if a context appears with every word. This gives the desired functionality; weights of relevant contexts are increased, all others decreased. At this point, it also becomes clear why *tf.idf* only counts the *qualitative* contextual appearances: The plain probability of a context word w_j co-occurring with the term w_i would express nothing about w_j's distribution over other contexts.

The second weighting scheme presented here is *pmi*, short for *pointwise mutual information*. *pmi* measures the association between a term w_i and a context word w_j, by calculating the probability that w_i and w_j occur together, divided by the probability that w_i and w_j are co-occurring by chance:

$$\begin{aligned} p(w_i, w_j) &= \frac{\mathbf{M}[i][j]}{\sum\limits_{i'=1}^{n} \sum\limits_{j'=1}^{n} \mathbf{M}[i'][j']} \\ p(w_i) &= \frac{\sum\limits_{j} \mathbf{M}[i][j]}{\sum\limits_{i'=1}^{n} \sum\limits_{j'=1}^{n} \mathbf{M}[i'][j']} \\ p(w_j) &= \frac{\sum\limits_{j} \mathbf{M}[i][j]}{\sum\limits_{i'=1}^{n} \sum\limits_{j'=1}^{n} \mathbf{M}[i'][j']} \\ pmi(w_i, w_j) &= \log\left(\frac{p(w_i, w_j)}{p(w_i) \cdot p(w_j)}\right) \end{aligned} \tag{3.13}$$

Again, the logarithm hints on its information-theoretical background: *pmi* measures the differing number of bits when encoding the joint event of w_i and w_j under the assumption that both w_i and w_j are independent. Hence, *pmi* becomes zero, if w_i and w_j are independent. When the probability of a co-occurrence of both terms is larger by chance than actually observed, the outcome is negative, and positive in case the observed probability of a co-occurrence of w_i and w_j is larger than under assumption

of independence.
In less technical terms, this means that a word-context pair receives a positive weight, if their co-occurrence is more probable than chance; otherwise, its weight becomes zero or below, if their appearance is exactly as or less probable than a random encounter.

3.2.1.2 Dimensionality Reduction

Up to this point, count-based models rely on the symbolic meaning of words. They cannot account for higher-level concepts, for instance *animateness* or *colour*, as long as these terms are not included. Even if these words are part of the model, the words - i.e., dimensions - would not be recognized as higher order abstraction, but rather as terms equally among others, only with differing weights. Another objection is that, although word vectors comprise high dimensionality, many entries remain zero (Turney and Pantel, 2010). These entries do - neither positively, nor negatively - contribute to the meaning of a term, and adversely affect the similarity between vectors. The fewer zero entries $\mathbf{M}$ has, the more the attraction and repulsion between terms is emphasized. Both can be accomplished through dimensionality reduction; this section presents how.
Starting with symmetric quadratic term-term matrices, the technique is extended to rectangular matrices. Due to its impact and widespread applications, it is covered in greater details. If not stated otherwise, the technicalities are based on chapters 2.1.5, 2.1.7 and 2.4 as well as theorem 8.1.1 of Golub and Loan (2013). However, this is only a introduction; for more details on properties and computations see Golub and Loan (2013).
Being so far viewed as a storage device for word vectors, the $n \times n$ term-term matrix $\mathbf{M}$ is now perceived as a linear operator from $\mathbb{R}^n$ to $\mathbb{R}^n$ that manipulates n-dimensional vectors through reflection and scaling[1]. The last operation is of particular interest; vectors being scaled by $\mathbf{M}$ are called *eigenvectors*. Formally, an eigenvector $\mathbf{v}$ fulfills the equation

$$\mathbf{M}\mathbf{v} = \lambda\mathbf{v} \tag{3.14}$$

with λ being a scalar, which is real for symmetric matrices. Any complex $n \times n$ matrix $\mathbf{A}$ with n linearly independent eigenvectors can be diagonalized. In this case, the eigenvectors form a basis for a new vector space, into which the operations can be reversibly projected:

$$\mathbf{A} = \mathbf{X}\mathbf{D}\mathbf{X}^{-1} \tag{3.15}$$

Here, $\mathbf{X}$ contains all eigenvectors as columns, and $\mathbf{D}$ is a diagonal matrix with the ith entry $\mathbf{D}[i][i]$ being set to the eigenvalue associated with the eigenvector in the ith

[1]Rotation and shearing are not provided by symmetric matrices. The general rotation matrix consists of alternating positive and negative sinus entries, thus cannot be symmetric (http://www.encyclopediaofmath.org/index.php?title=Rotation&oldid=11806 [Accessed: 6.8.2020]).
Shearing means to shift a vector along a particular axis, and is also not symmetric (http://www.encyclopediaofmath.org/index.php?title=Shear&oldid=40067 [Accessed: 6.8.2020]).

column of $\mathbf{X}$.

The benefit of decomposing a matrix is that axis along which vectors are scaled most (i.e., the eigenvectors) can be detected. Since the eigenvectors form the basis of a vector space (due to their assumed linear independence), one can define a subspace consisting of the eigenvectors with the largest associated eigenvalues, into which vectors can be projected. Let $\mathbf{X}_k$ contain the eigenvectors for the$k$ largest eigenvalues as columns,

$$\mathbf{A}_k = \mathbf{A}\mathbf{X}_k \tag{3.16}$$

gives the projection of $\mathbf{A}$ into the k-dimensional space spanned by the k largest eigenvectors. By this way, each word vector would obtain a low dimensional representation $\mathbf{A}_k \in \mathbb{R}^{n \times k}$.

In the general decomposition shown above, the eigenvectors might be complex. However, the subspace into which the real word vectors are projected should ideally be also real, for a better interpretability. Fortunately, in the case of real symmetric matrices as $\mathbf{M}$, eigenvalues and -vectors are real. Moreover, the eigenvectors are orthogonal, which makes the computation of the inverse much easier:

$$\mathbf{M} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T \tag{3.17}$$

By diagonalization, the orthogonal eigenvectors in $\mathbf{Q}$ can be turned orthonormal, by multiplying the associated eigenvalue with the inverse of the length. The inverse of each orthonormal matrix is its transposed. This diagonalization heavily facilitates the dimensionality reduction of the simplified symmetric term-term matrix.

With $\mathbf{M}$ being a general $n \times m$ matrix over $\mathbb{R}$, this is no longer possible. However, a useful technique, *singular value decomposition* (henceforth, SVD), allows to diagonalize also these kind of matrices. Forced to be symmetric by multiplying its transposed, it can be shown that there exists a singular value $\sigma \in \mathbb{R}$, such that

$$\begin{aligned} \mathbf{M}\mathbf{M}^T\mathbf{v} &= \sigma^2\mathbf{v} \\ \mathbf{M}^T\mathbf{M}\mathbf{u} &= \sigma^2\mathbf{u}. \end{aligned} \tag{3.18}$$

$\mathbf{v} \in \mathbb{R}^n$ is called the *right* and $\mathbf{u} \in \mathbb{R}^m$ the *left* singular vector. Based on this observation,

$$\begin{aligned} \mathbf{M}\mathbf{M}^T &= \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T \\ \mathbf{M}^T\mathbf{M} &= \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T, \end{aligned} \tag{3.19}$$

with $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$. Hence, the SVD of $\mathbf{M}$ is

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{3.20}$$

Thus, SVD is closely related to eigenvalue decomposition. Conveniently, the singular values in $\mathbf{\Sigma}$ are sorted decreasingly along the main diagonal; the remaining

entries are zero. Singular vectors in $\mathbf{U}$ and $\mathbf{V}$ are arranged accordingly.
There are two properties of SVD which should be highlighted here. The first one is of algebraic nature. Consider the projection of $\mathbf{M}$ into the subspace spanned by singular vectors with the top k singular values, in similar fashion as above:

$$\mathbf{M}_k = \mathbf{U}_k \Sigma_k \tag{3.21}$$

Then, $\mathbf{M}_k \in \mathbb{R}^{n \times k}$ is the closest approximation to $\mathbf{M}$ of rank and dimension k under Frobenius norm. This is ensured by the *Eckart-Young-Theorem* (Golub and Loan, 2013).
The second - geometric - property can be illustrated by the hyperdimensional ellipsoid E defined by

$$E = \{\mathbf{Mx} \mid \|\mathbf{x}\|_2 = 1\}, \tag{3.22}$$

i.e., a irregularly shaped n-dimensional sphere of diameter smaller or equal two. Here, the singular vectors in $\mathbf{U}$ describe the directions of the semi-axis, and the associated singular values their lengths. Illustratively, the singular vectors of the k largest singular values point into the direction of the most variance. As a desirable side effect, the approximated matrix becomes 'smoother', with reduced noise, and less zero entries (Turney and Pantel, 2010).
Deerwester et al. (1990) are among the first ones to use SVD in NLP to improve document queries, also coining the term *latent semantic analysis*. Since then, it finds widespread application, not only in distributional semantics, but also in unsupervised dictionary induction (cf. Section 4.2.2). Landauer and Dumais (1997) suggest that SVD can help to understand how language is acquired through induction from texts. They conclude, "[I]t is supposed that the co-occurrence of events, words in particular, in local contexts is generated by and reflects their similarity in some high-dimensional source space." Here, "high-dimensional source space" denotes the subspace the word vectors are projected into; due to its reduced noise, newly learned words can be categorized much faster.
Turney (2006) describe that LSA can be used to find relational similarity of the type w_1 *is to* w_2 *what* w_3 *is to* $w_?$. For example, the question could be, *mason* to *stone* is the same as *carpenter* to *?* (where the correct answer would be *wood*).
Another way to overcome sparse entries and define superficial concepts right from the start, is *random indexing* (RI) (Sahlgren, 2005), which is mentioned here for the sake of completeness. Avoiding the costly computation of singular values and -vectors, it assigns each word initially to a random, low-dimensional vector, whose entries only consist of minus one, zero, and plus one. This way, the vectors are *almost* orthogonal to each other, emulating a vector space. This dates back to the Johnson-Lindenstrauss-Lemma (Johnson and Lindenstrauss, 1984), which states that if data points are projected "into a randomly selected subspace of sufficiently high dimensionality, the distances between the points are approximately preserved" (Sahlgren, 2005). The word vectors are then constructed by adding the vector of a context word

to the vector of another term each time they co-occur in a text. Compared to SVD, this makes extending the model rather easy: If a new word ought to be included, a new random vector is created, the corpus is read again, and the vectors are added accordingly.
The amount to which the dimensionality is reduced depends on the method and the number of words, i.e. the original number of dimensions in the vector space. As the projected vectors in RI are not fully orthogonal to each other, more dimensions might be needed to reach the same expressiveness as LSA. Landauer and Dumais (1997) empirically determined around 300 dimensions to work best for 60,000 words in the English `TOEFL` synomymy tasks, with generally good results for values between 100 and 1,000 (however, "Computational constraints prevented assessing points above 1,050 dimensions" (Landauer and Dumais, 1997)), whereas Karlgren and Sahlgren (2001) use 1,800 RI-dimensions per 94,000 words for the same task. The maximum performance for LSA is 64.4% and for RI 72.0 % correct synonymy questions.
LSA and RI are intentionally presented in less detail, as the word vectors for this project are computed with a predictive model. which is introduced in the upcoming section.

3.2.2 Predictive Models

The methods presented so far are *count based*. Although the idea of neural language models is not new - Landauer and Dumais (1997) mention a neural network (NN) perspective on LSA - *predictive* models gain attention especially since the 2000s, enabled by advanced technology, most notably improved processors, graphical processing units and cloud computing. Starting with Bengio et al. (2001) and Bengio et al. (2003), network models lead to a first peak with Collobert and Weston (2008), aiming on a unified architecture for part-of-speech tagging, chunking, named entity recognition, semantic role labeling, language modeling, and synonymy tasks. In all these tasks, *embedded* word feature vectors, i.e. a latent semantic structure, abstract from a symbolic word representation to improve results. Words are initially represented as a one-hot encoding, meaning, each term in a vocabulary of size n is assigned to a zero-vector of size n, with exactly one entry being set to one. The extensive study of Baroni et al. (2014) suggests to prefer predictive models over count-based approaches in all accounts, especially in those relevant for this thesis: *relatedness* and *synonymy*.
This chapter presents two predominant models, WORD2VEC and GLOVE, as well as FASTTEXT, which additionally incorporates subword information.

3.2.2.1 WORD2VEC

The foundations of WORD2VEC (Mikolov et al., 2013) lie in (Mikolov et al., 2013b). In their recurrent neural network (RNN) language model, Mikolov et al. (2013b) predict the upcoming word ($\mathbf{y}(t)$) in a text by embedding the current one-hot-encoded word ($\mathbf{w}(t)$) and the embedded previous words ($\mathbf{s}(t)$) in a hidden layer:

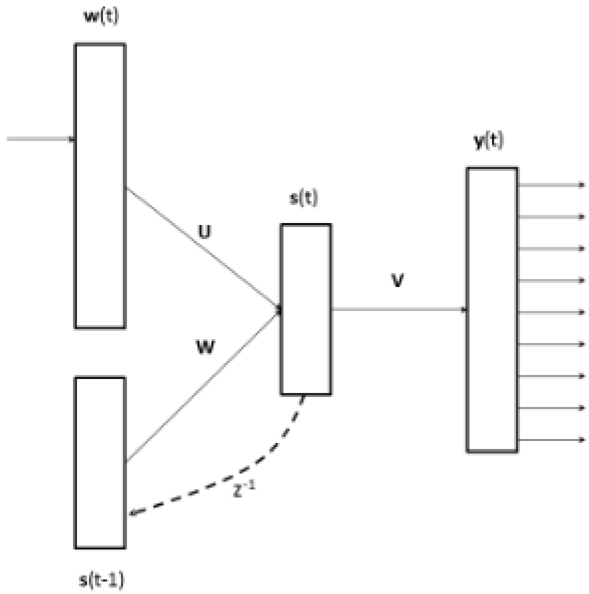


FIGURE 3.1: Architecture of the RNN used by Mikolov et al. (2013b)

$\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ are weight matrices. Formally, the layers are computed as follows:

$$\begin{aligned} \mathbf{s}(t) &= f(\mathbf{Uw}(t)) + \mathbf{Ws}(t-1) \\ \mathbf{y}(t) &= g(\mathbf{Vs}(t)), \end{aligned} \tag{3.23}$$

with

$$\begin{aligned} f(\mathbf{z}) &= \frac{1}{1+e^{-\mathbf{z}}} \\ g(\mathbf{z}[m]) &= \frac{e^{\mathbf{z}[m]}}{\sum\limits_{n} e^{\mathbf{z}[n]}} \end{aligned} \tag{3.24}$$

The word representation for the ith word is in the ith column of $\mathbf{U}$, and $\mathbf{y}$ is a distribution over all n words in the vocabulary. Using Mikolov's *RNN Toolkit*[2], which is based on his dissertation, the network is trained via stochastic gradient descent (SGD), specifically backpropagation through time, with certain extensions (Mikolov, 2012). A detailed description of SGD is given later in this section.

The main observation by Mikolov et al. (2013b) is that certain changes in linguistic features, such as switching gender (e.g., from *king* to *queen*) or grammatical number (for instance, from *king* to *kings*) are roughly represented as linear differences between the embedded word vectors.

[2] http://www.fit.vutbr.cz/~imikolov/rnnlm/ [Accessed: 6.8.2020]

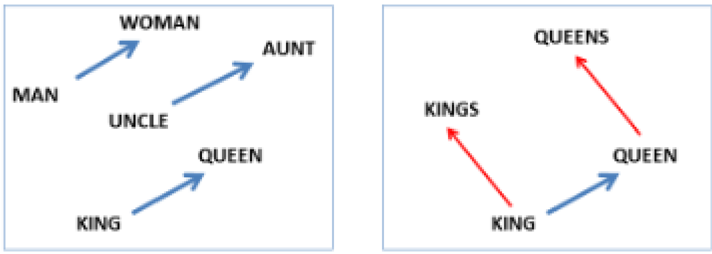


FIGURE 3.2: Sketch of gender and numerus relations (Mikolov et al., 2013b)

Now, the goal of WORD2VEC is to improve the results of this model. Instead of including an arbitrary number of previous words as in the RNN model, Mikolov et al. (2013) limit the context to a symmetric window. Experiments are conducted with two variants: In one case, a term is predicted by its context (continuous bag-of-words, CBOW), and in the other case vice versa (continuous skip-gram).

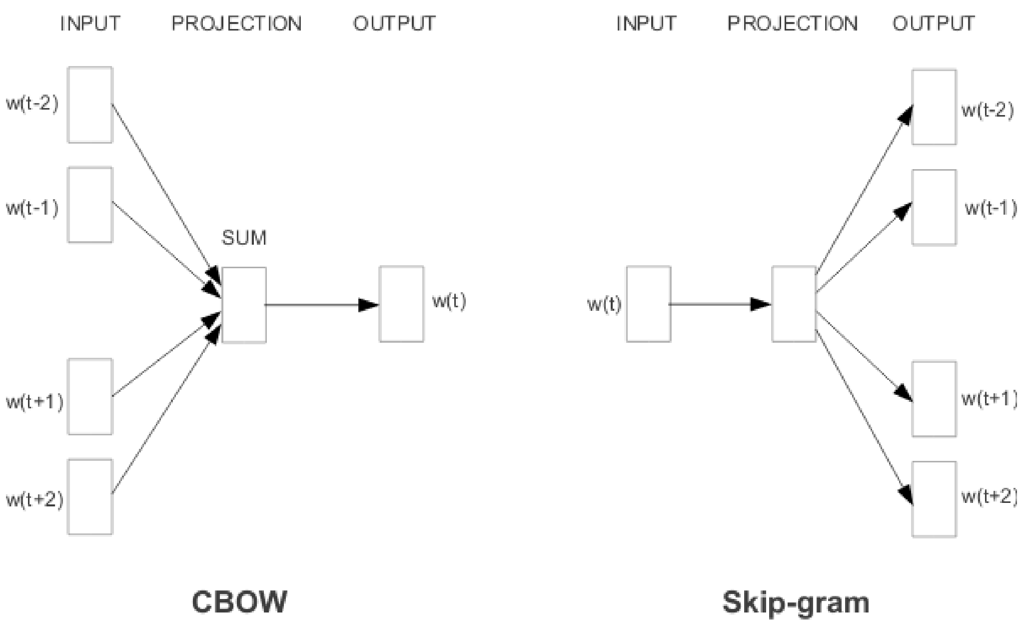


FIGURE 3.3: Overview over WORD2VEC architectures

$\mathbf{w}(t \pm x)$ denotes the word on xth position before or after the current word, $\mathbf{w}(t)$. In both architectures, the projection layer is the embedding vector for $\mathbf{w}(t)$. Besides changes in the architecture, the embedding level has a linear, the output layer a hierarchical softmax activation function with binary Huffman encoding. The reason for doing so is that, in the denominator, softmax loops over all possible words from the vocabulary. Considering a large vocabulary, this operation can become troublesome. With Huffman encoding, each word in the vocabulary V is converted into a binary code of optimal length, following these assumptions (Huffman, 1952):

V is finite

There exists a finite number $n = |V|$ of words $w_1 \ldots w_n \in V$.

The Probability Distribution over V is fully known

$P(w_1), \ldots, P(w_n)$ denotes the probability with which each word occurs in the data. Without loss of generality, let henceforth be $P(w_1) \geq \cdots \geq P(w_n)$.

No two words will consist of identical arrangements of coding digits.

Let b_x, b_y be the binary codes for words w_x, w_y. Then, the condition $w_x \neq w_y \Leftrightarrow b_x \neq b_y$ ought to hold.

No start or end marks

Once the beginning of a sequence is known, start or end marks should be irrelevant.

The length of the code of each word depends on its Probability

Let $L(w)$ be the length of word w. Then, $L(w_1) \leq L(w_2) \leq \cdots \leq L(w_{n-1}) = L(w_n)$ should hold: For a globally minimal code, frequent words need to be encoded with less signals than infrequent ones. The most infrequent word, w_n, has as many binary digits as w_{n-1}. That is, because any additional number of bits to encode specifically one word would be wasted.

Each binary substring of $L(n) - 1$ digits must occur as prefix or as an encoding on its own.

Otherwise, this substring would occur, again, specifically in the encoding of one of the two most infrequent words, and thus would be wasted.

Preserving these conditions, a binary tree is built by subsequently combining the two most infrequent words, i.e. nodes , in one node:

TABLE I

OPTIMUM BINARY CODING PROCEDURE

Message Probabilities												
Original Message Ensemble	Auxiliary Message Ensembles 1	2	3	4	5	6	7	8	9	10	11	12
												→1.00
											→0.60	
										→0.40	0.40	
									→0.36	0.36		
								→0.24	0.24	0.24		
0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20			
							→0.20	0.20	0.20			
0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18				
						→0.18	0.18	0.18				
					→0.14	0.14	0.14					
0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10					
0.10	0.10	0.10	0.10	0.10	0.10	0.10						
0.10	0.10	0.10	0.10	0.10	0.10	0.10						
				→0.10	0.10							
		→0.08	0.08	0.08	0.08							
			→0.08	0.08								
0.06	0.06	0.06	0.06	0.06								
0.06	0.06	0.06	0.06									
0.04	0.04	0.04	0.04									
*0.04	0.04	0.04										
0.04	0.04	0.04										
0.04	0.04											
	→0.04											
0.03												
0.01												

TABLE II

RESULTS OF OPTIMUM BINARY CODING PROCEDURE

i	$P(i)$	$L(i)$	$P(i)L(i)$	Code
1	0.20	2	0.40	10
2	0.18	3	0.54	000
3	0.10	3	0.30	011
4	0.10	3	0.30	110
5	0.10	3	0.30	111
6	0.06	4	0.24	0101
7	0.06	5	0.30	00100
8	0.04	5	0.20	00101
9	0.04	5	0.20	01000
10	0.04	5	0.20	01001
11	0.04	5	0.20	00110
12	0.03	6	0.18	001110
13	0.01	6	0.06	001111
			$L_{av}=3.42$	

TABLE 3.3: Exemplary Huffman-Encoding

In conclusion, Huffman encoding assigns shorter binary codes to frequent words. It can be shown that in the long run, the number of necessary computations is reduced to log(Unigram_Perplexity(V)), which turns out to be just the entropy of the vocabulary (see (Jelinek et al., 1977) and (Brown et al., 1992)). Note that this step is only necessary for the output. Since the input consists only of one-hot encoded vectors, the corresponding column vector can be easily retrieved, e.g. by hashing, which makes the costly matrix-vector multiplication obsolete.
Having encoded the vocabulary, each word is now decomposed into a series of 0-1-decisions in a binary (search) tree. In order to compute its probability, first, a function is needed that indicates which of both possibilities is further proceeded:

$$[\![x]\!] = \begin{cases} 1, \text{if } x = \texttt{True} \\ -1, \text{if } x = \texttt{False} \end{cases} \tag{3.25}$$

Next, each binary choice is assigned with a probability. Therefore, the position *within* the path/ sequence has to be determined. For every word w, $n(w,j)$ gives the jth position in the bit string of w, and $L(w)$ returns the length of that string. In this notation, $n(w,1)$ denotes the *root*, and $n(w,L(w))$ the *leaf* node w in the decision tree. Each of these positions $n(w,j)$ has its own feature vector, $\mathbf{v}_{n(w,j)}$. Let $ch(n)$

be a predetermined child (say, left) for any given node n. With that in mind, the probability of the transition from position $n(w,j)$ to the left child is

$$\begin{aligned} P(n(w,j+1) = \text{left} \mid \mathbf{v}_I) &= \sigma\left([\![\text{left} = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T \mathbf{v}_I\right) \\ &= \sigma\left(1 \cdot \mathbf{v}_{n(w,j)}^T \mathbf{v}_I\right), \end{aligned} \tag{3.26}$$

and similarly, the probability to traverse the right-hand side is given by

$$\begin{aligned} P(n(w,j+1) = \text{right} \mid \mathbf{v}_I) &= \sigma\left([\![\text{right} = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T \mathbf{v}_I\right) \\ &= \sigma\left(-1 \cdot \mathbf{v}_{n(w,j)}^T \mathbf{v}_I\right) \\ &= 1 - \sigma\left(\mathbf{v}_{n(w,j)}^T \mathbf{v}_I\right)) \\ &= 1 - P(n(w,j+1) = \text{left} \mid \mathbf{v}_I); \end{aligned} \tag{3.27}$$

for

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3.28}$$

and $\mathbf{v}_I$ an *input* vector. Either, $\mathbf{v}_I$ is an addition of the embedded context vectors (CBOW), or the single embedded vector of the center word (Skip-Gram). Finally, the probability of a predicted term w_O, given an embedded context or center vector $\mathbf{v}_I$ is calculated by the product of all binary decisions:

$$P(w_O \mid \mathbf{v}_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T \mathbf{v}_I\right)) \tag{3.29}$$

Computing $\log\left(P(w_O \mid \mathbf{v}_I)\right)$ and its gradient $\nabla \log\left(P(w_O \mid \mathbf{v}_I)\right)$ takes now only about $\mathcal{O}(L(w))$ operations. Applying the logarithm to probabilities is common as it preserves the maximum, and turns products into sums, which are less complicated to derive. In the following, the back propagation (BP) algorithm is described (cf. Rumelhart et al. (1988), chapter 6.5 in (Goodfellow et al., 2016), and Rong (2014)) and exemplarily executed for Skip-gram.

The basic idea behind BP is to propagate errors - meaning, the difference between the desired and the actual output of the network - throughout the layers. Weights are adjusted such that this difference is minimized. Iteratively, the gradient with respect to each weight is calculated, to find the direction of the steepest ascent. In order to descend to the global minimum, this direction of the gradient is then negatively pursued, with a certain step size, also called learning rate. The process comes to halt, when the output of the network converges. This does not necessarily mean that the global minimum has been reached; it could also be a local minimum, or a saddle point. Especially, the size of the steps needs to be carefully considered: Is it too small, the process does not converge in reasonable time; is it too large, the crucial minimum might be missed.

At this point the question arises, why the global minimum cannot be calculated right

away, by employing the first and second derivative. The reason is that the underlying function, which generates the input-output pairs is unknown, and therefore needs to be approximated. The term *stochastic* in SGD refers to the random initialization of the weight vectors. Figuratively, the gradient descent as just presented begins at some random starting point, and follows the path of steepest descent to a minimum.
Returning to WORD2VEC, first, a loss function E needs to be defined. As the probability in equation (3.26) ought to be maximized with respect to its parameters, the negative logarithm is a natural choice. If the loss E is minimized, it has the same effect as maximizing the probability.

$$\begin{aligned} E &= -log\left(\prod_{j=1}^{L(w)-1} \sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right)\right) \\ &= -\sum_{j=1}^{L(w)-1} \log\left(\sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right)\right) \end{aligned} \tag{3.30}$$

The next step is to compute how changes in the weight vectors $\mathbf{v}_{n(w,j)}$ and $\mathbf{v}_I$ influences E.

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{v}_{n(w,j)}^T\mathbf{v}_I} &= -\log\left(\sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right)\right)' \\ &= \frac{1}{\sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right)} \\ &\cdot \sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right)' \\ &= -\frac{1}{\sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right)} \\ &\cdot \sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right) \\ &\cdot \left(1-\sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right)\right) \\ &= -\left(1-\sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right)\right) \\ &= \sigma\left([\![n(w,j+1) = ch(n(w,j))]\!]\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right) - 1 \\ &= \begin{cases} \sigma\left(\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right) - 1, \text{if } n(w,j+1) = ch(n(w,j)) \\ -\sigma\left(\mathbf{v}_{n(w,j)}^T\mathbf{v}_I\right) \quad \text{else.} \end{cases} \end{aligned} \tag{3.31}$$

That is, because

$$\frac{\partial \log(x)}{\partial x} = \frac{1}{x} \tag{3.32}$$

and

$$\frac{\partial \sigma(x)}{\partial x} = (1-\sigma(x)) \cdot \sigma(x) \tag{3.33}$$

With these results, the specific partial derivatives can be easily obtained:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{v}_{n(w,j)}} &= \frac{\partial E}{\partial \mathbf{v}^T_{n(w,j)}\mathbf{v}_I} \cdot \frac{\partial \mathbf{v}^T_{n(w,j)}\mathbf{v}_I}{\partial \mathbf{v}_{n(w,j)}} \\ &= \frac{\partial E}{\partial \mathbf{v}^T_{n(w,j)}\mathbf{v}_I} \cdot \mathbf{v}_{n(w,j)} \\ &= \begin{cases} \left(\sigma\left(\mathbf{v}^T_{n(w,j)}\mathbf{v}_I\right) - 1\right) \cdot \mathbf{v}_I, \text{ if } n(w, j+1) = ch(n(w,j)) \\ -\sigma\left(\mathbf{v}^T_{n(w,j)}\mathbf{v}_I\right) \cdot \mathbf{v}_I \quad \text{else.} \end{cases}\end{aligned} \tag{3.34}$$

To get the partial derivative of $\mathbf{v}_I$, one has to sum over all possible $j = 1 \ldots L(w) - 1$:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{v}_I} &= \sum_{j=1}^{L(w)-1} \frac{\partial E}{\partial \mathbf{v}^T_{n(w,j)}\mathbf{v}_I} \cdot \frac{\partial \mathbf{v}^T_{n(w,j)}\mathbf{v}_I}{\partial \mathbf{v}_I} \\ &= \sum_{j=1}^{L(w)-1} \frac{\partial E}{\partial \mathbf{v}^T_{n(w,j)}\mathbf{v}_I} \cdot \mathbf{v}_{n(w,j)}.\end{aligned} \tag{3.35}$$

For Skip-Gram, this is done in every backward pass of a context word w.
Now, the update rules can be defined. For brevity, the derivations are displayed symbolically. Let η be the step size towards the steepest slope. Then,

$$\mathbf{v}^{t+1}_{n(w,j)} = \mathbf{v}^t_{n(w,j)} - \eta \left(\frac{\partial E}{\partial \mathbf{v}_{n(w,j)}}\right) \tag{3.36}$$

and

$$\mathbf{v}^{t+1}_{I_c} = \mathbf{v}^t_{n(w,j)} - \eta \left(\frac{\partial E}{\partial \mathbf{v}_I}\right) \tag{3.37}$$

denote the updated weight vectors at step $t+1$. This is the final step of in the parameter estimation of the Skip-gram model using hierarchical softmax. For CBOW, back-propagation functions analogously.
Furthermore, Mikolov et al. (2013a) give two extensions specifically for Skip-gram, which are briefly adressed here. *Negative Sampling* (NEG), which is based on *noise contrastive estimation* developed by Gutmann and Hyvärinen (2012), aims to train the word vectors by additionally providing negative examples. Not only should the model learn to predict the context words correctly, but also discriminate them from counterexamples. Therefore, in each prediction, k words are drawn from a noise distribution $P_n(w) = \frac{U(w)^{\frac{3}{4}}}{Z}$, where $U(w)$ is the unigram occurrence of w, and Z the total number of word occurrences:

$$\log\left(\sigma(\mathbf{v}^T_O\mathbf{v}_I)\right) + \sum_{O'=1}^{k} \mathbb{E}_{w_{O'} \sim P_n(w)} \left[\log\left(\sigma(-\mathbf{v}^C_{O'} BOWaT\mathbf{v}_I)\right)\right] \tag{3.38}$$

Depending on the size of the data set, k is chosen between 5 and 20 (for small sets), or between 2 and 5 (for large sets). The more positive examples can be presented to the model, the less negative counterexamples are necessary to enhance it. In less technical terms, the objective in equation (3.38) maximizes the probability of the actual context word w_O *plus* the counter-probability of k frequent, but non-present words $w_{O'}$. Mikolov et al. (2013a) admit that the design of P_n is not theoretically, but empirically justified.

The second adjustment to Skip-gram is *sub-sampling*. Functional words, such as determiners or prepositions, do not contribute to the meaning of a word. Being observably among the most frequent terms in the vocabulary, these words are discarded with probability

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}, \tag{3.39}$$

where w_i is a word from the vocabulary, $f(w_i)$ its frequency, and t a threshold. Mikolov et al. (2013) find that $t \approx 10^{-5}$ yields the best empirical results for a vocabulary size of 692,000 words:

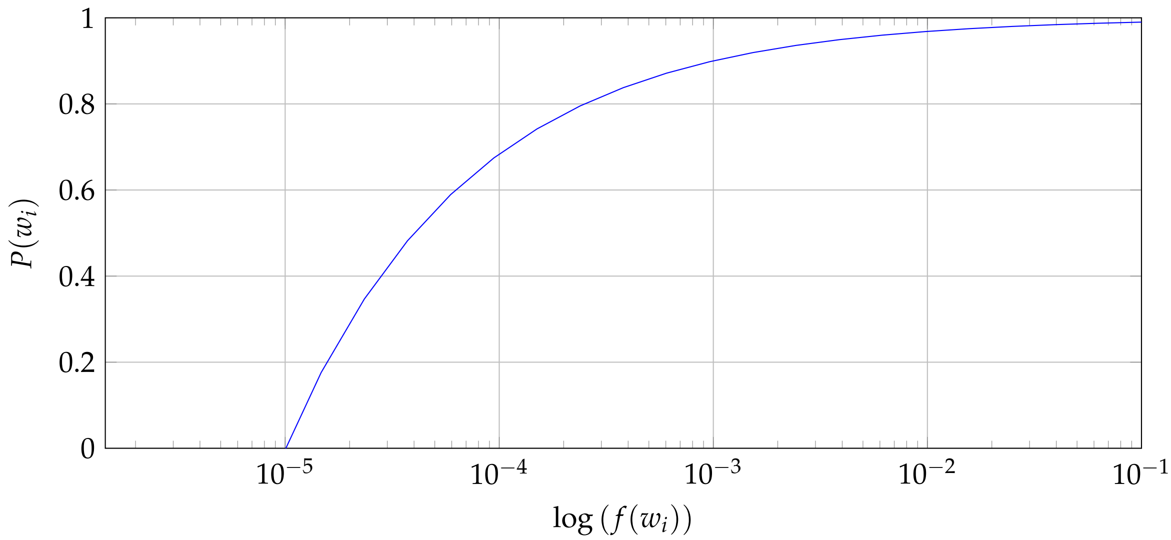


FIGURE 3.4: Plot of $P(w_i) = 1 - \sqrt{\frac{1}{10^5 \cdot f(w_i)}}$

After this introduction to WORD2VEC, the models' performances are presented. The main evaluation is based on two data sets; one measures syntactical, the other semantic understanding. Overall, 8,869 semantic and 10,675 syntactic questions are constructed in semi-supervised fashion: First, similar words are bundled in pairs, which are then automatically combined to form machine-readable questions. In both test settings, the system has to guess the closest word $w_?$ for w_1 *is to* w_2 *what* w_3 *is to* $w_?$. A syntactical type of question would be, '*quick* is to *quickly*, as *slow* is to ...?' , whereas the semantic questions ask, for example, '*Germany* is related to *Berlin* as *France* is related to ...?'. See Figure 3.8 for a complete overview about the questions types.

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

TABLE 3.4: Overview on Semantic and Syntactic Questions

The heuristic for selecting the missing words is already given in equations (3.9) and (3.10). The models are trained on `Google News` corpus. For better interpretability, the number of embedded vectors remains 30,000 during all experiments. The following table contains the results for CBOW and Skipgram, compared to the author's RNN and NN language model. In all these models, the word vectors are limited to 640 dimensions. The percentages refer to the number of correctly answered questions in ratio to all questions.

Model	Syntactic [%]	Semantic [%]
RNN Language Model	36	9
NN Language Model	53	23
CBOW	64	24
Skip-gram	59	55

TABLE 3.5: Overview of WORD2VEC-Results (Mikolov et al., 2013)

WORD2VEC clearly outperforms NN/RNN approaches of that time. While CBOW uses four words before and after each term as context, Skip-gram takes randomly between one and ten previous and forthcoming words. For more details on competing models and training specifics, see Mikolov et al. (2013).
To see the effects of NEG and subsampling in action, Mikolov et al. (2013a) evaluate the Skip-gram model with an embedding dimension of 300. NEG-5 refers to five, NEG-15 to fifteen counter-examples during each pass. Here, the context size was limited to five.

Method	Elapsed Time [min]	Syntactic [%]	Semantic [%]	Total Accuracy
NEG-5	38	63	54	59
NEG-15	97	63	58	**61**
Huffman	41	53	40	47

TABLE 3.6: Results for Skip-gram without Sub-Sampling (Mikolov et al., 2013a)

Method	Elapsed Time [min]	Syntactic [%]	Semantic [%]	Total Accuracy
NEG-5	14	61	58	60
NEG-15	26	61	61	**61**
Huffman	21	52	59	55

TABLE 3.7: Results for Skip-gram with Sub-Sampling ($t = 10^{-5}$) (Mikolov et al., 2013a)

As can be seen, NEG-5 accelerates training time, because the network converges faster when a small number of negative samples are provided. NEG-15, however, presents too many samples, such that training time decelerates. Both training instances provide better results than plain Huffman encoding.
Another important factor is dimensionality.

Dimensions\Training words	24M	49M	98M	196M	391M	783M
50	13.4	15.7	18.6	19.1	22.5	23.2
100	19.4	23.1	27.8	28.7	33.4	32.2
300	23.2	29.2	35.3	38.6	43.7	45.9
600	24.0	30.1	36.5	40.8	46.6	50.4

TABLE 3.8: Combined CBOW Accuracies [%] (Mikolov et al., 2013)

Table 3.8 shows the combined accuracies (in percentages) of CBOW for both syntactic and semantic questions. With increasing size of the corpus, the accuracy grows (with one exception). For smaller dimensions (50 and 100), the gain is lower than for larger embedding sizes (300 and 600). However, it is also evident that improvement for a fixed training size declines with growing vector sizes.
Conclusively, WORD2VEC can capture syntactic and semantic information better than previous approaches. Both CBOW and Skip-gram can be also used to learn phrase representations. This property is omitted here, because this project is only concerned with word-to-word translation.

3.2.2.2 GLOVE

Following the success of WORD2VEC, GLOVE (short for *global vectors*) aims to incorporate corpus statistics in the model (Pennington et al., 2014). While the former only uses global statistics implicitly, by sampling local contexts, the latter tries to factor the entries of the co-occurrence matrix constructed from a corpus, by low-dimensional word vectors. Their motivation for doing so is the observation from a corpus presented in the figure below:

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

TABLE 3.9: Sample Conditional Probabilities (Pennington et al., 2014)

The overview in Table 3.9 shows that sole conditional probabilities are not speaking for themselves. Only the relation to each other reveals their full expressiveness. Let w_i, w_j and w_k words, and $P(w_k \mid w_i)$ and $P(w_k \mid w_j)$ be the conditional probability of w_k being in the context of w_i, respectively w_j. The ratio of those conditional probabilities can now give information about the degree of association between w_k and w_i, w_j:

$$\frac{P(w_k \mid w_i)}{P(w_k \mid w_j)} = \begin{cases} \ll 1, \text{ if } w_k \text{ is } \textit{more} \text{ associated with } w_j \text{ than } w_i \\ \approx 1, \text{ if } w_k \text{ is } \textit{equally} \text{ (un)associated with } w_j \text{ and } w_i \\ \gg 1, \text{ if } w_k \text{ is } \textit{more} \text{ associated with } w_i \text{ than } w_j \end{cases} \tag{3.40}$$

An example for first case would be the second column in the figure above, where *gas* is more associated with *stream*, than it is with *ice*. *Water* and *fashion* are showcases the second instance, where *water* is, and *fashion* is not, both linked with *ice* and steam. And lastly, *solid* appears more often in the context of *ice*, than it does with *stream*. GLOVEs goal is to encode this reasoning in word vectors.
Starting with a conventional count-based distributional model, it is assumed that context vectors should not distinguish from regular term vectors; hence, **M** is both quadratic *and* symmetric. Each row (word) vector then becomes normalized by its sum, yielding each jth entry of the ith word to be the conditional probability $P(w_j \mid w_i)$. Some initially unspecified function F is now supposed to map word vectors $\mathbf{w}_i, \mathbf{w}_j$ and context vector $\tilde{\mathbf{w}}_k \in \mathbb{R}^d$ to those conditional probabilities:

$$F\left(\mathbf{w}_i, \mathbf{w}_j, \tilde{\mathbf{w}}_k\right) = \frac{\mathbf{M}[i][k]}{\mathbf{M}[j][k]} \tag{3.41}$$

In order to capture the relationships in vector spaces, the ratios of probabilities are translated into vector differences:

$$F\left(\mathbf{w}_i - \mathbf{w}_j, \tilde{\mathbf{w}}_k\right) = \frac{\mathbf{M}[i][k]}{\mathbf{M}[j][k]} \tag{3.42}$$

What is missing is the transformation from vectors on the right to a scalar on the left. As in the previous step, Pennington et al. (2014) prefer a linear approach by taking the scalar product:

$$\begin{aligned} F\left(\langle \mathbf{w}_i - \mathbf{w}_j, \tilde{\mathbf{w}}_k \rangle\right) &= F\left((\mathbf{w}_i - \mathbf{w}_j)^T \tilde{\mathbf{w}}_k\right) \\ &= F\left(\mathbf{w}_i^T \tilde{\mathbf{w}}_k - \mathbf{w}_j^T \tilde{\mathbf{w}}_k\right) \\ &= \frac{\mathbf{M}[i][k]}{\mathbf{M}[j][k]}. \end{aligned} \tag{3.43}$$

By changing *minuend* and *subtrahend* in the argument, enumerator and denominator should switch, too. This is reflected in the next equation:

$$F\left(\mathbf{w}_i^T \tilde{\mathbf{w}}_k - \mathbf{w}_j^T \tilde{\mathbf{w}}_k\right) = \frac{F\left(\mathbf{w}_i^T \tilde{\mathbf{w}}_k\right)}{F\left(\mathbf{w}_j^T \tilde{\mathbf{w}}_k\right)} \tag{3.44}$$

So far, an exponential function like $F(x) = e^x$ satisfies all equations so far:

$$e^{\left(\mathbf{w}_i^T \tilde{\mathbf{w}}_k\right)} \stackrel{!}{=} \frac{\mathbf{M}[i][j]}{\sum_{j'}^{n} \mathbf{M}[i][j']} \tag{3.45}$$

$$\mathbf{w}_i^T \tilde{\mathbf{w}}_k \stackrel{!}{=} \log(\mathbf{M}[i][j]) - \log(\sum_{j'}^{n} \mathbf{M}[i][j']) \tag{3.46}$$

However, word and context vectors are still not treated equally. Swapping $\mathbf{w}_i$ and $\tilde{\mathbf{w}}_k$ does not give the same result. This is why, $\log(\sum_{j'}^{n} \mathbf{M}[i][j'])$ is relocated to a general scalar bias b_{w_i}, and to finally restore symmetry, another bias $\tilde{b}_{w_k}$ is added:

$$\mathbf{w}_i^T \tilde{\mathbf{w}}_k + \tilde{b}_{w_k} + b_{w_i} \stackrel{!}{=} \log(\mathbf{M}[i][j]). \tag{3.47}$$

As noted before, co-occurrence matrices are very sparse. In order to prevent numerical errors emerging from $log(0)$, the authors propose a weighted least squares objective,

$$J = \sum_{i,j=1}^{n} f(\mathbf{M}[i][j]) \left(\mathbf{w}_i^T \tilde{\mathbf{w}}_j + b_{w_i} + \tilde{b}_{w_j} - \log(\mathbf{M}[i][j])\right)^2 \tag{3.48}$$

where $f(x)$ has to fulfill certain requirements:

Fast convergence

As x approaches zero, $\log^2(x)$ poses a numerical error. So, for small x, $f(x)$ should converge faster than $\log^2(x)$ diverges.

Non-decreasing

f is not supposed to overweight to rare co-occurrences.

Stagnand for large x

Furthermore, f should also not overemphasize frequent co-occurrences.

From the vast number of functions satisfying these three desiderata, Pennington et al. (2014) find that

$$f(x) = \begin{cases} {\frac{x}{x_{max}}}^{\frac{3}{4}}, \text{ if } x < x_{max}, \\ 1 \quad \text{otherwise} \end{cases} \tag{3.49}$$

yields the best empirical results, with x_{max} being fixed to 100. Mikolov et al. (2013a) discover a similar fractional power weighting in NEG (see plot 3.4).

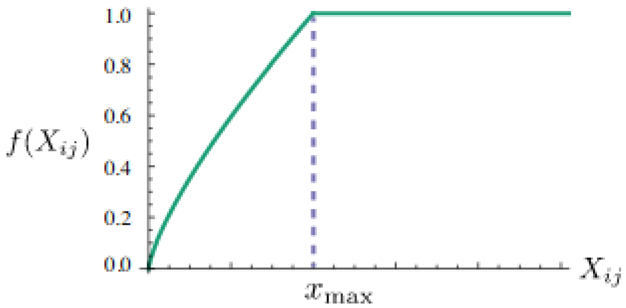


FIGURE 3.5: Plot of $f(x)$ (Pennington et al., 2014)

Objective J is minimized using ADAGRAD (Duchi et al., 2011), a variant of SGD. For convenience, J is multiplied by $\frac{1}{2}$, such that the exponent can be dropped while deriving. The gradients of the vectors and bias terms are given below:

$$\nabla_{\mathbf{w}_i} = f(x) \cdot \left(\mathbf{w}_i^T \tilde{\mathbf{w}}_j + b_{w_i} + \tilde{b}_{w_j} - \log(\mathbf{M}[i][j]) \right) \cdot \tilde{\mathbf{w}}_j \tag{3.50}$$

$$\nabla_{\tilde{\mathbf{w}}_j} = f(x) \cdot \left(\mathbf{w}_i^T \tilde{\mathbf{w}}_j + b_{w_i} + \tilde{b}_{w_j} - \log(\mathbf{M}[i][j]) \right) \cdot \mathbf{w}_i \tag{3.51}$$

$$\nabla_{b_{w_i}} = f(x) \cdot \left(\mathbf{w}_i^T \tilde{\mathbf{w}}_j + b_{w_i} + \tilde{b}_{w_j} - \log(\mathbf{M}[i][j]) \right) \cdot 1 \tag{3.52}$$

$$\nabla_{\tilde{b}_{w_j}} = f(x) \cdot \left(\mathbf{w}_i^T \tilde{\mathbf{w}}_j + b_{w_i} + \tilde{b}_{w_j} - \log(\mathbf{M}[i][j]) \right) \cdot 1 \tag{3.53}$$

Randomly initialized vectors and biases are then updated following ADAGRAD

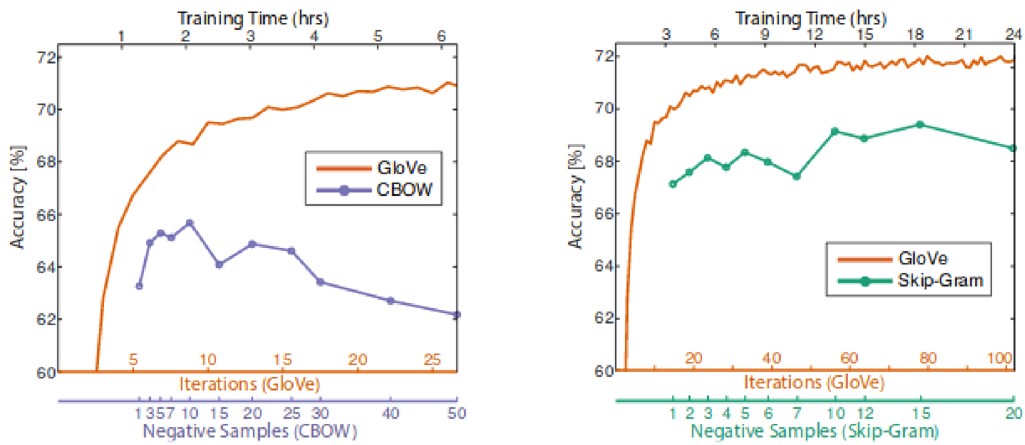
$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t - \frac{\eta}{\sqrt{\sum_{t'=1}^{t} \nabla_{\mathbf{w}_i^{t'}}}} \cdot \nabla_{\mathbf{w}_i^t} \tag{3.54}$$

$$\tilde{\mathbf{w}}_j^{t+1} = \tilde{\mathbf{w}}_j^t - \frac{\eta}{\sqrt{\sum_{t'=1}^{t} \nabla_{\tilde{\mathbf{w}}_j^{t'}}}} \cdot \nabla_{\tilde{\mathbf{w}}_j^t} \tag{3.55}$$

$$b_i^{t+1} = b_i^t - \frac{\eta}{\sqrt{\sum_{t'=1}^{t} \nabla_{b_i^{t'}}}} \cdot \nabla_{b_i^t} \tag{3.56}$$

$$\tilde{b}_j^{t+1} = \tilde{b}_j^t - \frac{\eta}{\sqrt{\sum_{t'=1}^{t} \nabla_{\tilde{b}_j^{t'}}}} \cdot \nabla_{\tilde{b}_j^t} \tag{3.57}$$

One benefit of this update rule is that the learning rate does not need to be adjusted during training, because it is divided by the root of the sum over the entries of all previous gradients. Figuratively, the step size decreases heavily, whenever the length of the gradient at time stamp t is large. In this case, the surface of the objective changes rapidly, and a small steps are advised, otherwise, the minimum might be missed. If the temporal gradient vector is comparable short, the step size does not reduce greatly, compared to the last step. Thus, as a second advantage, the resulting accuracy stabilizes much earlier than compared to regular SGD:



(a) GloVe vs CBOW (b) GloVe vs Skip-Gram

FIGURE 3.6: Comparison of Training Time/ Epochs between GLOVE and WORD2VEC

For a rigorous mathematical deduction of ADAGRAD, the the original paper is recommended.
Although there exist two vectors $\mathbf{w}_i$ and $\tilde{\mathbf{w}}_i$ for each word w_i, both differ only due to their random initialization, as long as $\mathbf{M}$ is symmetric. A small boost in terms of accuracy is detected, when both vectors are added.
Pennington et al. (2014) train GLOVE on two `Wikipedia` dumps with 1.0 and 1.6 billion tokens, the 4.3 billion tokens `Gigaword 5` corpus, a combination of `Wikipeda` and `Gigaword5` totalling to 6.0 billion tokens, and 42 billion tokens from web data. Each corpus is tokenized and set to lowercase, and the 400,000 most common words are used as vocabulary. Various context windows are tested; in order to decrease the influence of distant words, each context is weighted by the inverse of its distance d to the center term, $\frac{1}{d}$. Additionally, the impact of dimensionality is also investigated.
Two applications are especially important for this project: Word similarity and word analogy. Pennington et al. (2014) report also results on *named entity recognition*, however, this field is only of minor interest here, and is therefore omitted. To evaluate word similarity, five test sets are employed: `RG` (Rubenstein and Goodenough, 1965),

MC (Miller and Charles, 1991), WordSim-353 (Finkelstein et al., 2002), SCWS (Huang et al., 2012), and RW (Luong et al., 2013). A description of those data sets can be found below.

RG

Consisting of 65 word pairs "of ordinary English words", the RG set gives a human ranking between 4.0 (highest similarity) and 0.0 (no similarity at all) (Rubenstein and Goodenough, 1965).

MC

The MC data set uses 30 noun pairs from RG, more precisely 10 with the highest (3-4), intermediate (1-3), and lowest (0-1) similarity level. Their similarity is ranked by humans on a scale from 0-4.

WordSim-353

Finkelstein et al. (2002) implement a dataset consisting of 350 word pairs with human similarity ranking from 0 (totally unrelated) to 10 ("very much related or identical words").

SCWS

As Huang et al. (2012) note, test sets for word similarity tasks are often isolated, in the sense that the words, whose similarity should be rated, are presented out of context. The SCWS set is meant to tackle this downside. To get a broad variety, words are sampled based on their number of parts of speech, number of synsets in WordNet, and their frequency. For each word, two random, related synsets are retrieved from WordNet. Then, for each word, a sentence is selected from Wikipedia, if it matches the same parts of speech, as well as one or more of its synsets. Afterwards, the similarity of the overall 2,003 pairs $\langle$rare word, synset$\rangle$ are rated from 0-10 by human beings, while the Wikipedia sentences are presented simultaneously as additional source of information.

RW

RW is a collection of rare words with certain affixes, such as *un-* or *-ment*. For each of those words, two related synsets from WordNet are randomly selected, which are then rated by humans from 0 to 10. Overall, it comprises of 2,034 $\langle$rare word, synset$\rangle$ word pairs (Luong et al., 2013).

The performance of system can be calculated by its agreement with the manual rankings. First, the cosine similarity is computed between the word pairs in the test set, and ranked after their score. Analogously, the word pairs in the test sets are ranked likewise after their human-anotated score. Let $X_1 \dots X_n$ and $Y_1 \dots Y_n$ denote the ranks of n arbitrarily sorted word pairs determined by the system (X_i) and by humans (Y). Then, *Spearman's rank correlation* (Zar, 2005) is applied to both rankings:

$$\rho = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\left(\sum_{i=1}^{n} (X_i - \bar{X}) \cdot \sum_{i=1}^{n} (Y_i - \bar{Y})\right)}}, \tag{3.58}$$

with $\bar{X}$, $\bar{Y}$ being the mean ranks of X and Y. A perfect correlation would result in $\rho = 1$, whereas no correlation would yield $\rho = 0$.

Word analogies are evaluated on the same set of 19,544 questions with the same method as WORD2VEC, which is presented in the preceding section.

In the following, comparisons between GLOVE and three other approaches are drawn: SVD, CBOW, and Skip-Gram. All three competing models use a vocabulary of 400,000 most common words, just as GLOVE does. For baseline SVD, columns of the co-occurrence matrix **M** (which is also used as basis for GLOVE) are restricted to those of the 10,000 mostfrequent words. Two extensions, SVD-S and SVD-L, compute SVD after taking the square-root (SVD-S) or the logarithm (SVD-L) of the remaining entries. To avoid numerical errors in SVD-L, the entries of **M** are incremented by one beforehand. WORD2VEC is trained with a context window size of ten, and ten negative samples in NEG.

Table 3.10 shows the results on word similarity; all models come with vectors of 300 dimensions.

Model	Corpus	RG	MC	WordSim-353	SCWS	RW
SVD	6B	42.5	35.1	35.3	38.3	25.6
SVD-S	6B	71.0	71.5	56.5	53.6	34.7
SVD-L	6B	75.1	72.7	65.7	56.5	37.0
CBOW	6B	68.2	65.6	57.2	57.0	32.5
Skip-Gram	6B	69.7	65.2	62.8	58.1	37.2
GLOVE	6B	77.8	72.7	65.8	53.9	38.1
SVD-L	42B	74.1	76.4	74.0	58.3	39.9
GLOVE	42B	**82.9**	**83.6**	**75.9**	**59.6**	**47.8**
CBOW*	100B	75.4	79.6	68.4	59.4	45.5

TABLE 3.10: Spearman's $\rho \cdot 100$ for Different Data sets

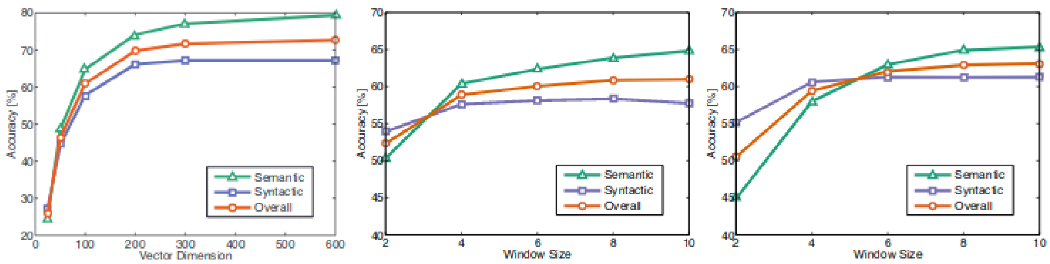
CBOW* is trained with word and phrase vectors on 100 billion news data. In most setups, GLOVEs results correlate best with human judgement, often with smaller corpora and vector sizes than other models. Especially the last two rows emphasize its potential, where only less than half of CBOWs corpora size is made use of to produce significant results.

The next table presents the accuracy on syntactic and semantic analogical questions. Evaluation is conducted in the same manner as WORD2VEC (Section 3.2.2.1).

Model	Dimension	Corpus	Semantic[%]	Syntactic[%]	Total Accuracy
GLOVE	100	1.6B	54.3	67.5	60.3
Skip-gram	300	1B	61	61	61
CBOW	300	1.6B	52.6	16.1	36.1
GLOVE	300	1.6B	61.5	80.8	70.3
SVD	300	6B	8.1	6.3	7.3
SVD-S	300	6B	46.6	36.7	42.1
SVD-L	300	6B	63.0	56.6	60.1
CBOW	300	6B	67.4	63.6	65.7
Skip-gram	300	6B	66.0	73.0	69.1
GLOVE	300	6B	67.0	77.4	71.7
CBOW	1000	6B	68.9	57.3	63.7
Skip-gram	1000	6B	65.1	66.1	65.6
SVD-L	300	42B	58.2	38.4	49.2
GLOVE	300	42B	**69.3**	**81.9**	**75.0**

TABLE 3.11: Results on Analogy Questions

GLOVE clearly outperforms the other approaches presented so far, often with smaller corpora and vector sizes. The authors note that the decrease in accuracy of SVD models for larger corpora showcases the necessity for weighting functions like f (Equation (3.49)).
The connection between vector/ window size and accuracy on word analogy tasks can be taken from the figure:



(a) Symmetric context (b) Symmetric context (c) Asymmetric context

FIGURE 3.7: Accuracy on Word Analogy Tasks depending on Vector and Context Size

Gains in accuracy with vector sizes above 200 are small. As syntactic information is encoded in close distance (for instance, through determiners), minor context sizes achieve better results on syntactic questions. Larger context windows work better for semantic questions, because relevant lexical terms, which contribute to the meaning of a word, “is more frequently non-local” (Pennington et al., 2014).
Similarly, asymmetric context windows capture syntactic relationships better, since asymmetry takes word order more into account.

3.2.2.3 FASTTEXT

All distributional methods so far treat words as symbolic occurrences in text. However, words themselves include again syntactic and semantic information, for example, grammatical number and gender, or subject and object marker, which are encoded in substrings. A whole research field in linguistics, *morphology*, is only concerned with how words are composed of subparts, *morphemes*. Therefore, it only seems straightforward to break words into into their units, to extract more syntactic and semantic information. As a motivating example, the words *dog* and *dogs* are treated up to now as two distinct words, whose similarity is only proven by the contexts they have in common.

Many approaches have been proposed to tackle this disadvantage. While many rely on annotated morphological data (for instance, Lazaridou et al. (2013) or Alexandrescu and Kirchhoff (2006)), only few models work fully unsupervised. Schütze (1993) employ SVD on character fourgrams. A word is thereby represented as the sum of all fourgram vectors surrounding it. For the purpose of language modeling, Mikolov et al. (2012) predict the upcoming word in a running text with a feed-forward NN based on previously encountered character n-grams. Luong et al. (2013) first conduct an unsupervised morphological segmentation before feeding the one-hot encoded segments into a RNN. Each word is then vectorized by the joint embeddings of its morphological subparts.

The method presented here, FASTTEXT (Bojanowski et al., 2017) is an advancement of the Skip-gram model with NEG. This is why, FASTTEXT is also referred to as *sisg*, or subword information Skip-gram. Recalling equation (3.38) for the Skip-gram model,

$$\log\left(\sigma(\mathbf{v}_O^T\mathbf{v}_I)\right) + \sum_{O'=1}^{k} \mathbb{E}_{w_{O'}\sim P_n(w)}\left[\log\left(\sigma(-\mathbf{v}_{O'}^T\mathbf{v}_I)\right)\right], \tag{3.59}$$

maximizing $\log(\sigma(x))$ is equivalent to

$$\begin{aligned}\max_x \log(\sigma(x)) &= \max_x \log(1) - \log\left(1+e^{-x}\right)\\ &= \max_x -\log\left(1+e^{-x}\right)\\ &= \min_x \log\left(1+e^{-x}\right)\end{aligned} \tag{3.60}$$

and maximizing $\log(\sigma(-x))$ gives similarly

$$\begin{aligned}\max_x \log(\sigma(-x)) &= \max_x \log\left(\frac{1}{1+e^{x}}\right)\\ &= \max_x \log(1)\log(1+e^{x})\\ &= \max_x -\log(1+e^{x})\\ &= \min_x \log(1+e^{x})\,.\end{aligned} \tag{3.61}$$

For convenience, Bojanowski et al. (2017) define a logistic loss function

$$\ell(x) = \log\left(1 + e^{-x}\right) \tag{3.62}$$

and rewrite (3.59) as

$$\ell\left(s(w_I, w_O)\right) + \sum_{O'=1}^{k} \mathbb{E}_{w_{O'} \sim P_n(w)} \left[\log\left(\ell\left(-s(w_I, w_{O'})\right)\right)\right] \tag{3.63}$$

where w_O is a context and w_I the center (and thus, the input) word. A more general setting loops over all possible center and context words w_t, w_c, and adds up their log-probabilities

$$\sum_{t=1}^{T} \left[\sum_{c} \ell\left(s(w_t, w_c)\right) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \log\left(\ell\left(-s(w_t, w_i)\right)\right) \right] \tag{3.64}$$

In the regular Skip-gram model, function $s(w_I, w_w)$ would evaluate to $\mathbf{v}_w^T \mathbf{v}_{w_I}$. However, since subword information is about to be incorporated, s is defined a little different.
First, a dictionary is constructed, which maps words to their character n-grams. In order to distinguish between pre-, post-, and infixes, start ($\langle$) and end tags ($\rangle$) are added to the front and to the back of each word. For instance, *where* is decomposed into the trigrams $\langle$*wh*, *whe*, *her*, *ere*, and *re*$\rangle$. Additionally, the whole word $\langle$*where*$\rangle$ is also included as a *special sequence* into the dictionary. The authors decide to use n-grams of length three to six. Let $G_w \subset \{1, \ldots, K\}$ be the set of n-grams plus the special sequence associated with a certain term w. Every n-gram g has a vector representation, $\mathbf{z}_g$. s is now modified such that

$$s(w_I, w_O) = \sum_{g \in G_{w_I}} \mathbf{z}_g^T \mathbf{v}_I, \tag{3.65}$$

meaning, w_I is decomposed into a sum of n-gram embeddings. The reason for s not being more complex is the observation that word relationships and analogies are linearly encoded in the vector space, which makes it possible to compose the meaning of a whole term from its parts. This formulation of s is then plugged into (3.64), and the resulting loss function minimized using SGD with a linearly decaying learning rate. How this looks like has been exemplarily shown for Skip-gram model in Section 3.2.2.1. To improve the efficiency, the n-grams are hashed to indices $1 \ldots K$ for faster access. In each pass, five negative examples are presented to the model, with a rejection threshold of 10^{-4} when sub-sampling the most frequent words. During all experiments, the embedding dimension is set to 300, and the context size uniformly sampled between one and five. For all languages, normalized `Wikipedia` articles serve as training corpus. Unfortunately, the authors do not go into detail about the training or vocabulary size, however, they note that words occurring less than five times are discarded during training. As Bojanowski et al. (2017) base their set-up on (Mikolov et al., 2013), it can be assumed that their vocabulary size is also in the vicinity of 30,000.

The evaluation is carried out in the same manner as in the previous sections. Both the agreement with manually graded word similarities, and the performance on word analogy tasks are measured. Besides English (`EN`), FASTTEXT is also tested on Arabic (`AR`), Czech (`CZ`), German (`DE`), Spanish (`ES`), Italian (`IT`), Romanian (`RO`) and Russian (`RU`). Word similarity is evaluated on the following data sets:

German

Word similarity in German is tested on `GUR65`, `GUR350` and `ZG222` (Gurevych (2005) and Zesch and Gurevych (2006)). `GUR65` is just a German translation of the `RG` data set by Rubenstein and Goodenough (1965). `GUR350` consists of nouns, verbs, and adjectives (Gurevych, 2005). `ZG222` comprises of 328 questions of parts-of-speech-tagged, lemmatized, and highly *tf.idf* weighted nouns, verbs, and adjectives from three German corpora (*BERUFEnet*, *German Indexing and Retrieval Testdatabase*, and *scientific PowerPoint presentations*). In all three data sets, the relatedness was ranked between zero (no relationship) and four (closely related).

English

The English data sets are the already presented `RW` by Luong et al. (2013) and `WordSim-353` from Finkelstein et al. (2002).

French

Analogously to `GUR65`, the French test set is a translation of `RG` (Joubarne and Inkpen, 2011).

Russian

The Russian `HJ` collection (Panchenko et al., 2016) is a Russian translation of the data sets `RG`, `MC` (Miller and Charles, 1991), and `WordSim-353`.

Arabic / Spanish / Romanian

Hassan and Mihalcea (2009) compose a dataset of `MC` and `WordSim-353` in Arabic, Spanish, and Romanian, ranked by human annotators from zero (unrelated) to four (synonymous).

Spearman's rank correlation results for the word similarity task are given below. In the case of sisg-, unknown words from the test sets are included as null-vector into the model. The regular sisg implementation rebuilds the words which are out of vocabulary, *OOV* for short, from its n-grams.

		Skip-gram	CBOW	sisg-	sisg
AR	WordSim-353	51	52	54	55
DE	GUR350	61	62	64	**70**
	GUR65	78	78	**81**	**81**
	ZG222	35	38	41	**44**
EN	RW	43	43	46	**47**
	WordSim-353	72	**73**	71	71
ES	WordSim-353	57	58	58	**59**
FR	RG	70	69	**75**	**75**
RO	WordSim-353	48	52	51	**54**
RU	HJ	59	60	60	**66**

TABLE 3.12: Spearman's $\rho \cdot 100$ of WORD2VEC and FASTTEXT on Word Similarity

First, one notices that FASTTEXT outperforms the baseline WORD2VEC systems. Also, rebuilding words which are OOV, is in all cases at least as good as representing them as null vectors. Second, the influence of subword information increases when dealing with languages with rich morphology, for instance German with its compounds and Russian, which has six grammatical cases. The large gap for the English `RW` data set (compared to `WordSim-353`) could be explained by its bias towards rare words, which are restored from n-grams, when OOV.
Analogies are tested on the following data sets:

Czech

Svoboda and Brychcin (2016) implement a compendium of 8,705 semantic and 13,552 syntactic questions. Specifically, the questions ask for: Presidents-states-cities (current presidents and capitals of European cities), antonyms, family relations (man-woman), gradation of adjectives (positive, comparative, and superlative), nationalities (feminine and masculine forms), nouns and their plural forms, job professions (feminine and masculine forms), verbs and their past forms, and pronouns in singular versus plural forms.

English

For English, Bojanowski et al. (2017) employ the same data set as Mikolov et al. (2013).

German

In order to test analogies in German, the data set from Köper et al. (2015) is used. It consists of German translations of the WORD2VEC data set, omitting the adjective-to-adverb questions, since there is no such destinction in German, and *paradigmatic* semantic relation questions. The latter ask for antonymy, synonymy, and hypernymy relationships and are crawled from `GermaNet`. Overall, there are 18,522 + 2,462 analogy questions in the data set.

Italian

The Italian analogical data set by Berardi et al. (2015) contains 19,791 questions, which are again translations of WORD2VECs compendium, with few adaptations to the comparative, superlative, some verb forms and feminine/ masculine singular and plural.

Table 3.13 shows accuracies for semantic and syntactic questions:

		Skip-gram	CBOW	sisg
CZ	Semantic	25.7	27.6	27.5
	Syntactic	52.8	55.0	77.8
DE	Semantic	66.5	66.8	62.3
	Syntactic	44.5	45.0	56.4
EN	Semantic	78.5	78.2	77.8
	Syntactic	70.1	69.9	74.9
IT	Semantic	52.3	54.7	52.3
	Syntactic	51.5	51.8	62.7

TABLE 3.13: Accuracies on Analogy Questions of WORD2VEC and FASTTEXT

Unsurprisingly, subword information improves the results on syntactic questions, especially on morphological rich data, such as German or Czech. On semantic analogies, the outcomes for Czech and English are still on par with those produced by WORD2VEC. However, for German and Italian, the results decrease, when compared to WORD2VEC. Bojanowski et al. (2017) note that this degradation is directly impacted by the choice of n-grams, as the next figure exhibits:

	2	3	4	5	6
2	57	64	67	69	69
3		65	68	70	70
4			70	70	**71**
5				69	**71**
6					70

(a) DE-GUR350

	2	3	4	5	6
2	59	55	56	59	60
3		60	58	60	62
4			62	62	63
5				64	64
6					**65**

(b) DE Semantic

	2	3	4	5	6
2	45	50	53	54	55
3		51	55	55	**56**
4			54	**56**	**56**
5				**56**	**56**
6					54

(c) DE Syntactic

	2	3	4	5	6
2	41	42	46	47	**48**
3		44	46	**48**	**48**
4			47	**48**	**48**
5				**48**	**48**
6					**48**

(d) EN-RW

	2	3	4	5	6
2	78	76	75	76	76
3		78	77	78	77
4			79	79	79
5				**80**	79
6					**80**

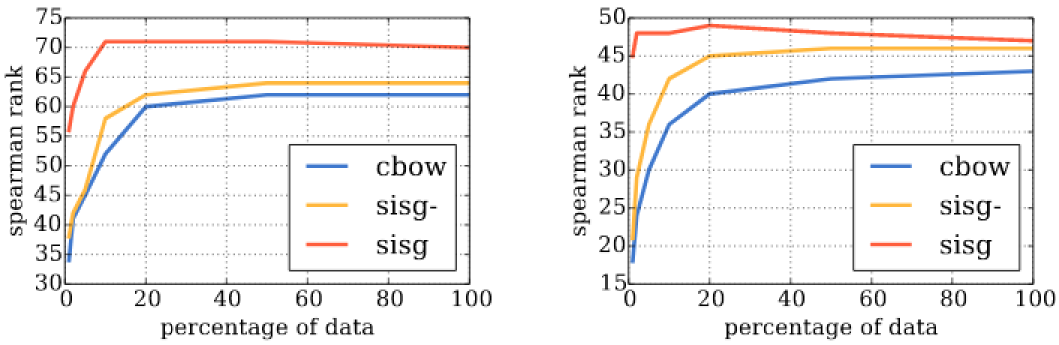
(e) EN Semantic

	2	3	4	5	6
2	70	71	73	74	73
3		72	74	**75**	74
4			74	**75**	**75**
5				74	74
6					72

(f) EN Syntactic

FIGURE 3.8: Effects of n-gram Sizes on German and English Analogies

In this experiment, word vectors are computed with n-grams ranging from i (row-value) to j (column value), and OOV words are restored from those n-grams. The initial choice of n being between three and six is empirically justified, as the results for two are unsatisfying, and the gains in accuracy between five and six are already diminishing. Also, the figure highlights that, for decent results on syntactic analogies, adding higher-order n-grams is sufficient, while for the performance on semantic questions, the lowest-order n-grams need to be subsequently removed with increasing an n. Otherwise, the broad prevalence of lower-order n-grams blurs the meaning of the words they are contained in.
However, because n-grams are always at least equally or more frequent than words, FASTTEXT consumes much less training data for the same results as WORD2VEC does:



(a) DE-GUR350 (b) EN-RW

FIGURE 3.9: Connection between Training Data Size and Performance on Word Similarity

Adding more training data, however, can decrease the performance, since the risk of diluting the meaning of n-grams by multiple contexts grows.
The practical effects of FASTTEXT can be exemplified best by a qualitative analysis. Figure 3.10 shows the most cosine-similar n-grams between a common word *chip* and an OOV word, *microcircuit*. Red nuances stand for high, blue ones for low similarity:

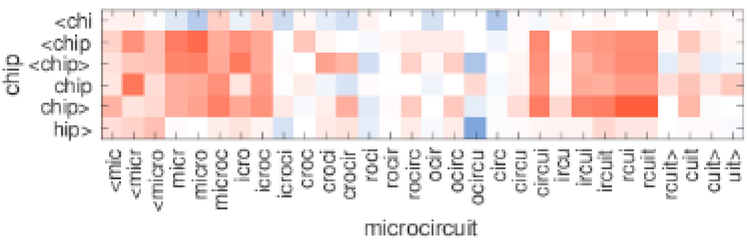


FIGURE 3.10: Plot of (Dis)similarity between n-grams of *chip* and *microcircuit* (OOV)

Especially those n-grams, which are clearly within the (sub-)words *micro*, *circuit*, and *chip* feature a high similarity; n-grams close to the boundaries, and in-between *micro* and *circuit* share a low similarity.

3.3 Proposed Method

The outstanding results of predictive models motivate their employment in MT architectures. Their ability to capture word similarities and analogies makes them predestined for translating terms in one language into another, where already small changes in meaning can have large effects (cf. idioms and collocations in Chapter 2). The method proposed in this thesis uses GLOVE with and without subword information. Reasons for preferring GLOVE over WORD2VEC are, on the one hand, its improved results on the aforementioned tasks, and, on the other hand, the explicit usage of global corpora statistics. The hypothesis is that global statistics help to detect associated terms over different languages, through of correlating frequency ranks in similar domains. Subword information is additionally included, as it is expected to facilitate the proper translation of declinated or conjugated words.
However, instead of building word meanings from n-grams, the outlined proposal uses states from finite-state automata (henceforth, FSA). Finite-state techniques have a long-standing history in NLP (for one of the first applications, see (Chomsky, 1956)), with focus on phonology (cf. (Kaplan and Kay, 1994) for a good overview) and morphology (such as (Beesley and Karttunen, 2003), for a general introduction with a wide variety of examples). Formally, an FSA is defined by a quintuple

$$\mathcal{A} = (Q, \Sigma, \delta, q_0, F) \tag{3.66}$$

where Q is a set of all states, Σ is an alphabet, δ is a transition function

$$\delta : \; Q \times \Sigma \; \mapsto \; Q, \tag{3.67}$$

q_0 a start state and F the set final states (Hopcroft et al. (2001), Definition 2.2.1). An FSA can be viewed as dictionary, which stores words along labeled paths. Every path begins at start state q_0, and ends in one of the final states in F. Function δ defines, which states are reachable from any given state, by consuming the upcoming symbol in the word. In practical applications, symbols are equal to single characters. Theoretically, a symbol could be also a sequence of characters; however, this would blow-up the alphabet to an unreasonable size. The automaton reads a word letter by letter, guiding the remaining string from one state to another. If a word w is fully processed, and the lastly visited state is in F, w is said to be *accepted*. Analogously, if the last state is not in F, or there is no subsequent state for the upcoming character in w according to δ, w is not accepted. Empty, so-called ϵ-transitions, where no symbol is processed to get to the next state, are not considered here. Any state q cannot be left by the empty string, i.e. $\delta(q, \epsilon) = q$.

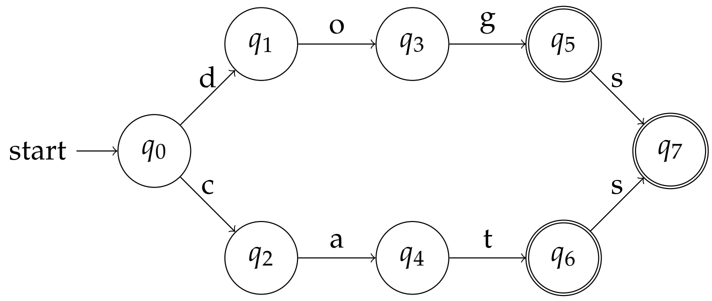


FIGURE 3.11: Examplary FSA

Figure 3.11 shows an exemplary FSA $\mathcal{A}$, with $Q = \{q_0, \ldots, q_6\}, \Sigma = \{a, c, d, g, o, s, t\}$, $\delta(q_0, d) = q_1$, $\delta(q_0, c) = q_2$, $\delta(q_1, o) = q_3$, $\delta(q_2, a) = q_4$, $\delta(q_3, g) = q_5$, $\delta(q_4, t) = q_5$, and $\delta(q_5, s) = q_6$. The start state is denoted by q_0, and the set of final states is $F = \{q_5, q_6\}$. $L(\mathcal{A}) = \{dog, dogs, cat, cats\}$.

Making this description more formal, let $\hat{\delta} : Q \times \Sigma^* \mapsto Q$, be an extension to δ which takes strings as argument, and $w = c_1 c_2 \ldots c_n \subset \Sigma^n$ a word consisting of n arbitrary characters from the alphabet:

$$\begin{aligned}
\hat{\delta}(q_0, w) &= \hat{\delta}(\delta(q_0, c_1), c_2 \ldots c_n) \\
&= \hat{\delta}(\delta(\delta(q_0, c_1), c_2), c_3 \ldots c_n) \\
&= \ldots \\
&= \delta(\hat{\delta}(q_0, c_1 c_2 \ldots c_{n-1}), c_n)
\end{aligned} \tag{3.68}$$

To be fully consistent, if δ is undefined for some state q and a symbol a, the word is directed to a non-accepting state $q_\perp$, with $q_\perp \notin F$, and $\delta(q_\perp, a) = q_\perp, \forall a \in \Sigma$.

The language $L(\mathcal{A})$ is then the set of words which reach a final state (Hopcroft et al. (2001), section 2.2.5):

$$L(\mathcal{A}) = \{w \mid \hat{\delta}(q_0, w) \in F\}. \tag{3.69}$$

$\mathcal{A}$ is called *deterministic*, if δ returns *exactly one* state for each input pair $\langle$state, symbol$\rangle$. That means, all outgoing transitions from a certain state have a distinct symbol (Hopcroft et al. (2001), Definition 2.3.2). Every non-deterministic FSA can be transferred into a deterministic one (Hopcroft et al. (2001), chapter 2.3.5), which accepts the same language. The benefit of determinized automata is that the set of strings by which each state is traversed is unique. Determinism is one of three important properties which are exploited later on.

The second feature is *minimization* (Hopcroft et al. (2001), section 4.4.3). For every deterministic FSA $\mathcal{A}$, there exists an equivalent minimal $\mathcal{A}'$, such that

$$L(\mathcal{A}) = L(\mathcal{A}') \text{ and} \tag{3.70}$$

$$\|Q\| \geq \|Q'\|. \tag{3.71}$$

In minimized FSAs, states with the same incoming and outgoing transitions are merged. That means, if two strings share a certain substring of length larger or equal two, they pass the same state(s). So, without any morphological annotation, words are categorized according to their pre-, suf-, and infixes This unsupervised classification into general units is thought to improve results in similarity, analogy, and MT.

The third essential notion, *acyclic*, rather concerns the topology of the underlying graph, than the language of the automaton. As the goal is to use the states of an FSA instead of symbolic words, every word should be uniquely represented by a finite number of states it runs through. Moreover, the order of the states should not matter; this way, the word could be encoded as a vector $\mathbf{v}$ of dimension $\|Q\|$, where the entries stand for the states in the automaton. If state $q_i \in Q$ is traversed, $\mathbf{v}[i] = 1$, otherwise $\mathbf{v}[i] = 0$. For instance, the word *dog* from the examplary FSA would be encoded by vector $\mathbf{v}_{dog} = \begin{pmatrix}1 & 1 & 0 & 1 & 0 & 1 & 0 & 0\end{pmatrix}$, as it traverses states q_0, q_1, q_3 and q_5. However, this is only possible, if the underlying graph contains no cycles: Assume an FSA $\mathcal{A}$, whose directed graph is acyclic, and a word w which passes some path of states, $q_0, \ldots, q_i, \ldots q_j, \ldots, q_n$. If w would not be distinctively represented by state indices $0, \ldots, i, \ldots, j, \ldots, n$, there has to be a second word, w', which passes exactly the same states, just in different order. Without loss of generality, let $q_0, \ldots, q_j, \ldots, q_i, \ldots, q_n$ be the path of states for w'. In this case, there has to be a set of transitions leading from q_i to q_j, as well as from q_j to q_i. This poses a contradiction to the assumption of $\mathcal{A}$ containing no cycles. Ergo, w is represented uniquely by the set of states it passes.

In order to construct a automaton for a given set of words that is deterministic, minimal, and acyclic, the algorithm of Daciuk et al. (2000) is applied. The algorithm incrementally builds and minimizes a deterministic FSA from an alphabetically sorted set of words[3]. Incrementality is crucial, because the number of states could grow exponentially without constant minimization. It uses an alternative definition of minimality, which can be shown to be equivalent with the one above:

$$\text{Minimal}(\mathcal{A}) \equiv \left(\underset{q,q' \in Q}{\forall} : q \neq q' \Rightarrow \vec{L}(q) \neq \vec{L}(q') \right) \land \text{Reachable}(\mathcal{A}) \tag{3.72}$$

$\vec{L}$ is denotes the *right* language of a state; this comprises of all substrings leading to an accepting state:

$$\vec{L}(q) = \{x \in \Sigma^\star \mid \hat{\delta}(q, x) \in F\} \tag{3.73}$$

Reachable($\mathcal{A}$) is a binary function which evaluates to `True`, if all states in $\mathcal{A}$ are reachable from the start state, otherwise to `False`:

$$\text{Reachable}(\mathcal{A}) \equiv \underset{q \in Q}{\forall} \underset{x \in \Sigma^\star}{\exists} : \hat{\delta}(q_0, x) = q. \tag{3.74}$$

Thus, if each state is reachable and has a unique right language, $\mathcal{A}$ is minimal. A

[3]The set of words is initially not in alphabetical order. But the algorithm for an unordered set is more complicated to implement than sorting the vocabulary

crucial part is to identify equal states. Of course, one could compute the right language of all states, and merge those with coinciding sets of words. However, such an approach is very costly. Daciuk et al. (2000) elaborate four 'local' criteria, which have to be fulfilled for two states q, q' to be equal:

1. Both are either final or non-final, and

2. have the same number of outgoing transitions, with

3. the same labels, which

4. lead to the same states.

If one of the conditions is not met, both q, q' are not equivalent and stored in a separate register.
The procedure of the algorithm is now as follows: Input words are in ascending alphabetical order. Every time a new word is added to the automaton, it is checked whether there exists a partial path, which accepts some prefix of the word. If the last state of this path (which could be, in case of now common prefix, the start state) has any subsequent states, all lastly appended states (the ones to which the alphabetically highest labeled[4] arcs point to) are revised, according to the four criteria. That is, because these successor states are not going to be visited by any other word, due to the alphabetical order of the input words. During revision, the state is either deleted, with all arcs pointing to them being redirected to an equivalent state, if such an identical state exists, or registered as new state. Equivalence could also be tested later, but doing so would increase the number of states exponentially in worst case, before minimization. In the next step, a new path of states is added to the automaton, which encodes the remaining substring of the current word.
If all words are processed, the states, to which the most recently introduced transitions (again, those with the alphabetically highest labels) point to, are revised.
In the course of the execution, each state has to be *replaced* (by an existing equivalent) or *registered* (into the set of states) only once. This operation costs $\mathcal{O}(\log n)$, both for searching the set of states, and possibly inserting a new one, when binary search is applied. This is done as often as letters are in the input words, namely l times. Thus, the overall run-time is $\mathcal{O}(l \log n)$.

```
Register := ∅;
do there is another word →
    Word := next word in lexicographic order;
    CommonPrefix := common_prefix(Word);
    LastState := δ*(q0, CommonPrefix);
    CurrentSuffix := Word[length(CommonPrefix)+1...length(Word)];
    if has_children(LastState) →
        replace_or_register(LastState)
    fi;
    add_suffix(LastState, CurrentSuffix)
od;
replace_or_register(q0)
```

```
func common_prefix(Word) →
    return the longest prefix w of Word such that δ*(q0, w) ≠ ⊥
cnuf

func replace_or_register(State) →
    Child := last_child(State);
    if has_children(Child) →
        replace_or_register(Child)
    fi;
    if ∃q∈Q(q ∈ Register ∧ q ≡ Child) →
        last_child(State) := q: (q ∈ Register ∧ q ≡ Child);
        delete(Child)
    else
        Register := Register ∪ {Child}
    fi
cnuf
```

FIGURE 3.12: Main Function (left) and Helper Methods (right)

[4]Hereby, the first rank within the alphabetical order is meant, starting with a, and ending with z.

Figure 3.12 presents the algorithm from Daciuk et al. (2000) as explained above. Method *last_child* returns the most recently added child state of the input state. Function *replace_or_register* searches for equivalent states among those lastly appended successors of the input state, and deletes them if necessary. The common prefix, which is already accepted by the automaton, is computed by *common_prefix*, while *add_suffix* introduces new states which accept the remaining chunk of *Word*.
In order to save memory and computation time, not all states are considered. Recalling the FSA from Figure 3.11, the word *dog* is represented by vector
$\mathbf{v}_{dog} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$, whereas q_5 would be solely sufficient to describe the path properly. Once the path towards q_1 is taken, subsequent states are determined. In this sense, the vector dimension of seven can be reduced to three: q_5, q_6, and q_7 are the only relevant states, needed to encode the four words. The set of *relevant* states is formalized in the next equation:

$$\text{Relevant}(Q) = \left\{ q \in Q \middle| q \in F \vee |\{q' \mid \exists a \in \Sigma : \delta(q', a) \stackrel{!}{=} q\}| \geq 2 \right\} \tag{3.75}$$

This definition serves the intuition from the example: A relevant state is either final, or has two or more preceding states. If it would only have one predecessor, it is predetermined to be reached.
These relevant states now become row- and column vectors of the co-occurrence matrix for GLOVE. Every time a word from the dictionary is encountered in the context of another one, the corresponding entries of the states which are passed are increased by one.
For a better readability, the evaluation procedure and the results for the acquired embedding vectors and the dictionary induction is compactly presented together (see Chapter 5). The next chapter describes how dictionaries can be induced automatically in first place.

Chapter 4

Dictionary Induction

"Thus one is led to the concept of a translation process in which, in determining meaning for a word, account is taken of the immediate (2N word) context."
Warren Weaver (Weaver, 1955)

Having obtained a vector representation for words in one language, the question is how to establish an alignment between vectors from different languages. One main problem is that the *meaning* behind individual dimensions in the vectors is arbitrarily determined while their construction. Therefore, conventional distance functions between vectors of different languages are not applicable, even if they correspond in the number of dimensions.
Approaches to dictionary induction can be roughly categorized into three groups, depending on their perspective on the representation of word meaning. Word-to-word (respectively word-to-embedding) matrices can either be viewed as probabilistic distributions, data points in a high-dimensional spaces, or (bipartite) graphs with weighted edges. In the first case, similarities between distributions are exploited. From the analytical point of view, the goal is to explicitly minimize the distance between translated source words and target words. In case of graphical approaches, the aim is to determine the similarity of two vertices from different graphs recursively by the similarity of their neighbors.
Some methods make use of a seed dictionary, based on which conclusions on further bilingual relationships are drawn.

4.1 Probabilistic Approaches

Probabilistic methods are historically the first experiments to semi- und unsupervised construction of dictionaries. Notable approaches are here Rapp (1995) and Rapp (1999). The starting point for both studies are word-co-ocurrence matrices. Rapp (1995) experiments with a modified mutual information to weight important context words and uses a matrix distance to find a permutation of row and column vectors that minimizes the distance between both co-occurrence matrices. Rapp (1999) employs log-likelihood ratios to find highly associated contexts and a vector distance to identify the most corresponding word vector in the other language. Koehn and Knight (2002) extends this approach of plain contextual information by

additionally utilizing string similarity and word frequency.
Based on their proof of concept, this section presents two more elaborate techniques, using canonical correlation analysis and generative adversarial nets.

4.1.1 Canonical Correlation Analysis

Developed by Hotelling (1936), canonical correlation analysis (henceforth abbreviated CCA) aims to find "basis vectors for two sets of variables such that the correlation between the projections of the variables onto these basis vectors is mutually maximized" (Hardoon et al., 2004). The correlation between two vectors of equal dimensions $\mathbf{u}$, $\mathbf{v}$ coincides in this context with the cosine similarity:

$$corr(\mathbf{u},\mathbf{v}) = \frac{\langle \mathbf{u},\mathbf{v}\rangle}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2} = cos(\mathbf{u},\mathbf{v}) \tag{4.1}$$

$\langle \cdot, \cdot \rangle$ denotes the inner product, and can be rewritten in matrix terms as

$$\langle \mathbf{u},\mathbf{v}\rangle = \mathbf{u}^T\mathbf{v} \tag{4.2}$$

Let now $\mathbf{S}_x$ and $\mathbf{S}_y$ be two samples comprising multivariate random vectors $\mathbf{x}_1 \dots \mathbf{x}_n$ and $\mathbf{y}_1 \dots \mathbf{x}_n$. $\mathbf{x}$- and $\mathbf{y}$-vectors can be of arbitrary dimensions, with $\mathbf{x}_{1\dots n} \in \mathbb{R}^{d_x}$ and $\mathbf{y}_{1\dots n} \in \mathbb{R}^{d_y}$. $\mathbf{x}_i$ can, but does not have to, correspond with $\mathbf{y}_i$. Thus, $\mathbf{x}_{1\dots n}$ and $\mathbf{y}_{1\dots n}$ are linearly projected onto a new direction by vectors $\mathbf{w}_x \in \mathbb{R}^{d_x}$ and $\mathbf{w}_y \in \mathbb{R}^{d_y}$:

$$\mathbf{x} \mapsto \langle \mathbf{w}_x, \mathbf{x}\rangle \tag{4.3}$$

$$\mathbf{y} \mapsto \langle \mathbf{w}_y, \mathbf{y}\rangle \tag{4.4}$$

The linear operators $\mathbf{w}_x$, $\mathbf{w}_y$ are chosen such that

$$\max_{\mathbf{w}_x,\mathbf{w}_y} corr(\mathbf{S}_x\mathbf{w}_x, \mathbf{S}_y\mathbf{w}_y) = \frac{\langle \mathbf{S}_x\mathbf{w}_x, \mathbf{S}_y\mathbf{w}_y\rangle}{\|\mathbf{S}_x\mathbf{w}_x\| \, \|_2\mathbf{S}_y\mathbf{w}_y\|_2} \tag{4.5}$$

is maximized. As the length of $\mathbf{w}_x$,$\mathbf{w}_y$ is levelled out by the denominator, their vector norm can be set to one without loss of generality. Uurtio et al. (2018) rephrase the introductory quotation more technically: "In summary, the principle behind CCA is to find two positions in the two data spaces respectively that have images on a unit ball such that the angle between them is minimised and consequently the canonical correlation is maximised."
The problem of maximizing the formula from above can be solved by the generalized eigenvalue problem (see (Hardoon et al., 2004) and (Uurtio et al., 2018)). Therefore, the variables in $\mathbf{S}_{x,y}$ are assumed to be zero-centered. Let

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_{xx} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{C}_{yy} \end{pmatrix} \tag{4.6}$$

be the joint covariance matrix of $\mathbf{S}_{x,y}$, with

$$\begin{aligned} C_{xx} &= \frac{1}{n}\mathbf{S}_x^T\mathbf{S}_x \\ C_{xy} &= \frac{1}{n}\mathbf{S}_x^T\mathbf{S}_y = C_{xy}^T \\ C_{yy} &= \frac{1}{n}\mathbf{S}_y^T\mathbf{S}_y \end{aligned} \tag{4.7}$$

Now, equation (4.5) can be reformulated in matrix notation as

$$\max_{\mathbf{w}_x,\mathbf{w}_y} corr(\mathbf{S}_x\mathbf{w}_x, \mathbf{S}_y\mathbf{w}_y) = \frac{\mathbf{w}_x^T\mathbf{C}_{xy}\mathbf{w}_y}{\sqrt{\mathbf{w}_x^T\mathbf{S}_{xx}\mathbf{w}_x}\sqrt{\mathbf{w}_y^T\mathbf{S}_{yy}\mathbf{w}_y}} \tag{4.8}$$

As mentioned before, $corr(\mathbf{x}, \mathbf{y})$ is invariant to any scaling of $\mathbf{x}$ and $\mathbf{y}$. Hence, maximizing (4.8) means to find the maximum for the numerator

$$\max_{\mathbf{w}_x,\mathbf{w}_y} f(\mathbf{w}_x, \mathbf{w}_y) = \mathbf{w}_x^T\mathbf{C}_{xy}\mathbf{w}_y \tag{4.9}$$

under the conditions

$$g_1(\mathbf{w}_x) = \mathbf{w}_x^T\mathbf{S}_{xx}\mathbf{w}_x - 1 \stackrel{!}{=} 0 \tag{4.10}$$

$$g_2(\mathbf{w}_y) = \mathbf{w}_y^T\mathbf{S}_{yy}\mathbf{w}_y - 1 \stackrel{!}{=} 0 \tag{4.11}$$

Constrained optimization problems can be solved using *Lagrange multipliers*. Following Kalman (2009) in notation and explanation, both objective and constraint functions need an image in $\mathbb{R}$, meaning they should map arguments onto one real number, and have to be continuously differentiable, which holds true for f, g_1, and g_2. In order to find the maximum, let the set of permissible vectors $\mathbf{w}_x$, $\mathbf{w}_y$ be given by function $r(t) = (\mathbf{w}_x, \mathbf{w}_y)$. r can be interpreted as an inverse function, which returns the input arguments for a given outcome t. If multiple arguments $(\mathbf{w}_x, \mathbf{w}_y)$ which produce the same result t, one arbitrary input pair can be fixed. The maximum is denoted as point $(\mathbf{w}_x^\star, \mathbf{w}_y^\star)$, with value $t^\star$. It is evident that $g_1(r(t)) = g_2(r(t)) \stackrel{!}{=} 0$ is constant, and that $f(r(t))$ has a maximum at $t^\star$. Therefore, the partial derivations of f, g_1 and g_2 into the direction of $r(t)$ become zero at $t^\star$, hence their gradients are perpendicular on $r(t)$ and parallel in point $t^\star$. Thus, the maximum $t^\star$ is where

$$\begin{aligned} \nabla f \stackrel{!}{=} \nabla g_1 \stackrel{!}{=} \nabla g_2 \Leftrightarrow \\ \nabla f - \nabla g_1 - \nabla g_2 \stackrel{!}{=} 0 \end{aligned} \tag{4.12}$$

holds. Only the magnitude of the gradients might differ, and it is unclear whether they point into the same or opposing directions. This is why, the constraints are assigned to scalars $\lambda_x, \lambda_y \in \mathbb{R}$, the so called Lagrange multipliers. The combined

Lagrangian of (4.9), (4.10) and (4.11) is then

$$\begin{aligned} L(\lambda_x, \lambda_y, \mathbf{w}_x, \mathbf{w}_y) &= f(\mathbf{w}_x, \mathbf{w}_y) - \lambda_x \cdot g_1(\mathbf{w}_x) - \lambda_y \cdot g_2(\mathbf{w}_y) \\ &= \mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y - \lambda_x (\mathbf{w}_x^T \mathbf{S}_{xx} \mathbf{w}_x - 1) - \lambda_y (\mathbf{w}_y^T \mathbf{S}_{yy} \mathbf{w}_y - 1) \end{aligned} \tag{4.13}$$

In the next step, the partial derivatives of $L(\lambda_x, \lambda_y, \mathbf{w}_x, \mathbf{w}_y)$ are calculated and set to zero:

$$\frac{\partial L}{\partial \mathbf{w}_x} = \mathbf{C}_{xy} \mathbf{w}_y - 2\lambda_x \mathbf{C}_{xx} \mathbf{w}_x \stackrel{!}{=} 0 \tag{4.14}$$

$$\frac{\partial L}{\partial \mathbf{w}_y} = \mathbf{C}_{yx} \mathbf{w}_x - 2\lambda_y \mathbf{C}_{yy} \mathbf{w}_y \stackrel{!}{=} 0 \tag{4.15}$$

Many publications divide $\lambda_{x,y}$ by two, such that there are no bothersome scalars left in the derivatives. Usually, one would have to compute also $\frac{\partial L}{\partial \lambda_x}$ and $\frac{\partial L}{\partial \lambda_y}$ to solve the equation system. However, it is possible to exploit the constraints in (4.10) and (4.11) to bypass this step: Multiplying $\frac{\partial L}{\partial \mathbf{w}_x}$ and $\frac{\partial L}{\partial \mathbf{w}_y}$ with $\mathbf{w}_x^T$ and $\mathbf{w}_y^T$ from the left gives

$$\begin{aligned} \mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y - 2\lambda_x \mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x \stackrel{!}{=} 0 \Leftrightarrow \\ \mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y - 2\lambda_x \cdot 1 \stackrel{!}{=} 0 \end{aligned} \tag{4.16}$$

and

$$\begin{aligned} \mathbf{w}_y^T \mathbf{C}_{yx} \mathbf{w}_x - 2\lambda_y \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y \stackrel{!}{=} 0 \Leftrightarrow \\ \mathbf{w}_y^T \mathbf{C}_{yx} \mathbf{w}_x - 2\lambda_y \cdot 1 \stackrel{!}{=} 0 \end{aligned} \tag{4.17}$$

Thus, $\lambda_x = \lambda_y$, and can be subsumed under a single λ. Plugging this into equation (4.14) yields

$$\begin{aligned} \mathbf{C}_{xy} \mathbf{w}_y - 2\lambda \mathbf{C}_{xx} \mathbf{w}_x &\stackrel{!}{=} 0 \Leftrightarrow \\ \mathbf{C}_{xy} \mathbf{w}_y &\stackrel{!}{=} 2\lambda \mathbf{C}_{xx} \mathbf{w}_x \rightarrow \\ \frac{\mathbf{C}_{xy} \mathbf{w}_y}{2\lambda} &\stackrel{!}{=} \mathbf{C}_{xx} \mathbf{w}_x \Leftrightarrow \\ \mathbf{w}_x &\stackrel{!}{=} \frac{\mathbf{C}_{xx}^{-1} \mathbf{C}_{xy} \mathbf{w}_y}{2\lambda}, \end{aligned} \tag{4.18}$$

and rewriting $\mathbf{w}_x$ accordingly in (4.15) results in

$$\begin{aligned} \frac{\mathbf{C}_{yx} \mathbf{C}_{xx}^{-1} \mathbf{C}_{xy} \mathbf{w}_y}{2\lambda} - 2\lambda \mathbf{C}_{yy} \mathbf{w}_y \stackrel{!}{=} 0 \Leftrightarrow \\ \mathbf{C}_{yy}^{-1} \mathbf{C}_{yx} \mathbf{C}_{xx}^{-1} \mathbf{C}_{xy} \mathbf{w}_y \stackrel{!}{=} 4\lambda^2 \mathbf{w}_y \end{aligned} \tag{4.19}$$

which can be solved by the standard eigenvalue problem (Uurtio et al., 2018). In case $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$ are not invertible, the equations can be reformulated by the generalized eigenvalue problem, $\mathbf{Ax} = \lambda \mathbf{Bx}$ (Hardoon et al., 2004). In this sense, CCA can be also perceived as finding general latent concepts (eigenvectors), in which two data sets $\mathbf{S}_x$ and $\mathbf{S}_y$ are maximally correlated. The degree of correlation correspondes with the square root of the eigenvalues λ. The vectors $\mathbf{w}_{x_i}$ and $\mathbf{w}_{y_i}$ for ith largest λ $\mathbf{w}_x$ maximize the correlation among concepts which remain uncorrelated by the previous $i-1$ $\mathbf{w}_x, \mathbf{w}_y$ canonical pairs. At most, there are $d = min(d_x, d_y)$ canonical correlations (Bach and Jordan, 2005). Vectors from $\mathbf{S}_x$ and $\mathbf{S}_y$ can now be projected into this d-dimensional subspace, in which they are maximally correlated, by arranging all $\mathbf{w}_{x_i}$ as column vectors in a matrix $\mathbf{U}_x$, and analogously all $\mathbf{w}_{y_i}$ in a matrix $\mathbf{U}_y$. Bach and Jordan (2005) further show that CCA can be used to calculate parameters in maximum-likelihood estimations for latent variables. This property is now exploited in the approach by Haghighi et al. (2008). For the formal proof, interested readers are directed to Bach and Jordan (2005), as it would exceed the scope of the thesis.

In their study, word vectors in the source and target language are thought to be connected by a latent, language-independent feature vector. Feature vectors of a word $s_i \in S$ in the source and $t_j \in T$ in the target language are denoted by $f_S(s_i)$, and $f_T(t_j)$, respectively. The translation process starts by assigning a random matching between a source and target word. There, every word is mapped either at *one* or *none* counterpart. Each matching between words s_i and t_j is assumed to be connected over a d-dimensional latent concept, $\mathbf{z}_{ij}$, which is drawn from a normal distribution with zero mean and variance one:

$$z_{ij} \sim \mathcal{N}(0, \mathbf{I}_d) \tag{4.20}$$

The source feature vector $f_S(s_i)$ then need to be generated by a normal distribution, with mean $\mathbf{W}_S\mathbf{z}_{ij}$ and variance $\mathbf{\Psi}_S$ to explain language-specific variations:

$$f_S(s_i) \sim \mathcal{N}(\mathbf{W}_S\mathbf{z}_{ij}, \mathbf{\Psi}_S), \tag{4.21}$$

where $\mathbf{W}_S \in \mathbb{R}^{d_S \times d}$. Similarly,

$$f_T(t_j) \sim \mathcal{N}(\mathbf{W}_T\mathbf{z}_{ij}, \mathbf{\Psi}_T), \tag{4.22}$$

with $\mathbf{W}_T \in \mathbb{R}^{d_T \times d}$. Both matrices $\mathbf{W}$ can be conceived as linear manipulations of the the shared concept's multi-dimensional mean. However, as Haghighi et al. (2008) note, they do "not play an explicit role in inference". Unmatched terms in source and target are drawn from a normal distribution with variance $\sigma^2\mathbf{I}_{d_{S,T}}$, with the variance being $\sigma^2 \gg 0$, as yet unmatched terms are viewed to be far off the mean of the included features.

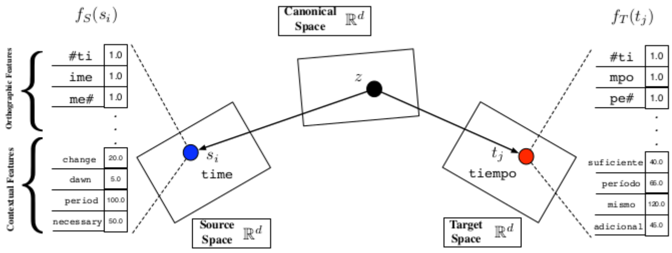


FIGURE 4.1: Illustration by Haghighi et al. (2008)

In order to find the best matching $m \in M$, the log-likelihood of all matches m between source and target word types S and T, given the parameter $\theta = \{\mathbf{W}_S, \mathbf{\Psi}_S, \mathbf{W}_T, \mathbf{\Psi}_T\}$, needs to maximized. Applying the logarithm to probabilities facilitates the calculation, because the multiplication can be transformed to a summation. Due to the strictly monotone behavior of the logarithmic function, the location of any maxima or minima remains the same, with only their magnitudes changing:

$$l(\theta) = \log p(S, T; \theta) = \log \sum_{m \in M} p(m, S, T; \theta) \tag{4.23}$$

In an iterative process, the parameters are first adjusted to maximimze the above log-likelihood, and afterwards, the *expected* matching is calculated on the basis of these parameters. Based on that updated matching, the parameters are again adjusted as in the first step. This approach is called Expectation-Maximization algorithm (Dempster et al., 1977).
In the maximization-step, CCA is finally applied to maximize

$$\max_{\theta} \log \sum_{(i,j) \in m} p(s_i, t_j; \theta) \tag{4.24}$$

CCA finds the matrices $\mathbf{U}_S \in \mathbb{R}^{d_S \times d}$ and $\mathbf{U}_T \in \mathbb{R}^{d_T \times d}$, which project source and target feature vectors into a subspace $\mathbb{R}^d$, where the vectors are maximally correlated. Bach and Jordan (2005) prove that the parameters θ can be then calculated as follows:

$$\begin{aligned} \mathbf{W}_S &= \mathbf{C}_{SS} \mathbf{U}_S \mathbf{P}^{\frac{1}{2}} \\ \mathbf{\Psi}_S &= \mathbf{C}_{SS} - \mathbf{W}_S \mathbf{W}_S^T \\ \mathbf{W}_T &= \mathbf{C}_{TT} \mathbf{U}_T \mathbf{P}^{\frac{1}{2}} \\ \mathbf{\Psi}_T &= \mathbf{C}_{TT} - \mathbf{W}_T \mathbf{W}_T^T \end{aligned} \tag{4.25}$$

where $\mathbf{P} \in \mathbb{R}^{d \times d}$ contains the canonical correlations and

$$\begin{aligned} \mathbf{C}_{SS} &= \frac{1}{m} \sum_{(i,j)} f_S(s_i) f_S(s_j)^T \\ \mathbf{C}_{TT} &= \frac{1}{|m|} \sum_{(i,j) \in m} f_T(t_i) f_T(t_j)^T \end{aligned} \tag{4.26}$$

are the empirical correlation matrices of the feature vectors being currently matched. In the expectation-step, the (bipartite) matching m with the highest associated weight is calculated. That is, because the classical computation would require to consider *all* possible matchings m':

$$m = arg \max_{m'} logp(m', S, T; \theta) \tag{4.27}$$

To transfer the matching optimization into maximizing the weights of a bipartite graph, pointwise mutual information (see (3.13) is applied, to measure the association between a source-target pair s_i, t_j:

$$\begin{aligned} w_{ij} &= log \frac{p(S, T; \theta)}{p(s_i; \theta) \cdot p(t_j; \theta)} \\ &= logp(s_i, t_j; \theta) - log(p(s_i; \theta) \cdot p(t_j; \theta)) \\ &= logp(s_i, t_j; \theta) - (logp(s_i; \theta) + logp(t_j; \theta)) \\ &= logp(s_i, t_j; \theta) - logp(s_i; \theta) - logp(t_j; \theta) \end{aligned} \tag{4.28}$$

It can be shown that the construction of the weights leads to the objective function defined above, only with some additional constant c:

$$\log p(m, S, T; \theta) = \sum_{(i,j) \in m} = w_{ij} + \mathsf{c} \tag{4.29}$$

Weights which are negative are set to zero. Haghighi et al. (2008) use then the Hungarian Algorithm (Kuhn, 1955) to compute the maximal assignment in the bipartite graph. The algorithm itself is not too difficult, though quite lengthy, thus readers are referred to Kuhn (1955). By the definition of weights w_{ij}, the maximal matching corresponds with the maximization of the log-likelihood.
Iteratively, the Expectation-Maximization-Algorithm establishes this way a maximum bipartite matching between the source and target language words.

4.1.2 Generative Adversarial Nets

First introduced by Goodfellow et al. (2014), the idea behind generative adversarial nets "is to set up a game between two players." One player, called *generator*, aims to imitate samples from the training distribution, which the other player, the *discriminator*, then classifies as to whether the given sample is generated or part of the training set (Goodfellow (2016), page 17-18). Both generator and discriminator try to minimize cost functions according to their goals. By doing so, the generator approaches stepwise the unknown and incomplete distribution of the input data.
Conneau et al. (2017) apply this idea to unsupervised translation. First, $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ ($\{\mathbf{y}_1, \ldots \mathbf{y}_m\}$) embedding vectors are obtained from FASTTEXT. Then, the generator is initialized as random matrix $\mathbf{W} \in \mathbb{R}^{d_y \times d_x}$ which maps the vectors from the d_x-dimensional source embedding space to the d_y-dimensional target embedding

space. The discriminator's task is now to distinguish between sampled projections $\mathbf{W}\mathbf{x}_i$ and actual target word vectors $\mathbf{y}_j$.
Let θ_D be the parameters of the discriminator and $P_\theta(\text{source} = 1 \mid \mathbf{z})$ the probability, with which, under parameters θ_D, a given vector $\mathbf{z}$ is a projected source embedding vector. Then,

$$\mathcal{L}_D(\theta_D \mid \mathbf{W}) = -\frac{1}{n}\sum_{i=1}^{n} \log P_\theta(\text{source} = 1 \mid \mathbf{W}\mathbf{x}_i) - \frac{1}{m}\sum_{i=1}^{m} \log P_\theta(\text{source} = 0 \mid \mathbf{y}_i) \tag{4.30}$$

denotes the loss-function of the discriminator that is to be minimized.
It can be derived by the following steps: Let

$$\bar{\mathcal{L}}_D(\theta_D \mid \mathbf{W}) = \left(\prod_{i=1}^{n} P_\theta(\text{source} = 1 \mid \mathbf{W}\mathbf{x}_i)\right)^{\frac{1}{n}} \cdot \left(\prod_{i=1}^{m} P_\theta(\text{source} = 0 \mid \mathbf{y}_i)\right)^{\frac{1}{m}} \tag{4.31}$$

the modified cost-function for the discriminator, which has to be maximized. The first factor calculates the geometric mean of the probability that all mapped source embeddings are correctly identified, while the second factor computes the mean of the probability that *all* target embeddings are successfully distinguished.
Next, the log-likelihood of the probability is calculated:

$$\log \bar{\mathcal{L}}_D(\theta_D \mid \mathbf{W}) = \log \left(\prod_{i=1}^{n} P_\theta(\text{source} = 1 \mid \mathbf{W}\mathbf{x}_i)\right)^{\frac{1}{n}} + \log \left(\prod_{i=1}^{m} P_\theta(\text{source} = 0 \mid \mathbf{y}_i)\right)^{\frac{1}{m}} \tag{4.32}$$

By the rules of logarithmic calculation, exponents become factors:

$$\log \bar{\mathcal{L}}_D(\theta_D \mid \mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} \log P_\theta(\text{source} = 1 \mid \mathbf{W}\mathbf{x}_i) + \frac{1}{m}\sum_{i=1}^{m} \log P_\theta(\text{source} = 0 \mid \mathbf{y}_i) \tag{4.33}$$

In order to apply SGD (as shown in section 3.2.3.1), both summands have to be negated for the global minimum to be found. By doing so, equation (4.30) is obtained. Feeding very rare words into the discriminator poses a disadvantage, because highly infrequent terms may not bear language characteristics, by which they could be distinguished from generated word vectors. This is why, only common words, being uniformly sampled, are used as input for the discriminator.
Analogously, only with opposing motives, the objective for generator $\mathbf{W}$ is deduced:

$$\mathcal{L}_W(\mathbf{W} \mid \theta_D) = -\frac{1}{n}\sum_{i=1}^{n} \log P_\theta(\text{source} = 0 \mid \mathbf{W}\mathbf{x}_i) - \frac{1}{m}\sum_{i=1}^{m} \log P_\theta(\text{source} = 1 \mid \mathbf{y}_i) \tag{4.34}$$

During training, $\mathbf{W}$ is ensured to be (almost) orthogonal. This has several benefits, as the dot-product and the ℓ_2 distance between vectors are retained and characteristics of the monolingual embeddings can be better translated into another embedding space. The update-rule of Cisse et al. (2017) imposes an orthogonal regularization on **W**. Let $\beta \in \mathbb{R}$ be the penalty for **W** not being orthogonal. Negative β encourage non-orthogonality, zero means that this feature is ignored, and positive β promote orthogonality. One attribute of orthogonal matrices is that multiplication with its transposed yields the identity matrix. So, an appropriate regularizer for **W** could look like this:

$$R_\beta(\mathbf{W}) = \frac{\beta}{2} \left\| \mathbf{W}\mathbf{W}^T - \mathbf{I} \right\|_2^2 \tag{4.35}$$

The **W** which minimizes the regularization above is now of special interest. Therefore, R_β is derived partially into the direction of **W**. As seen before, dividing by two is just a convenience, to get rid of the exponent. This gives

$$\nabla_{\mathbf{W}} R_\beta(\mathbf{W}) = \beta(\mathbf{W}\mathbf{W}^T - \mathbf{I})\mathbf{W} \tag{4.36}$$

In every step, **W** should move a towards orthogonality. Equation (4.36) gives the direction of the steepest ascend. As it needs to be minimized, $\nabla_{\mathbf{W}}$ is negated:

$$-\nabla_{\mathbf{W}} R_\beta(\mathbf{W}) = \beta\mathbf{W} - \beta\mathbf{W}\mathbf{W}^T\mathbf{W} \tag{4.37}$$

W is updated by adding it to $-\nabla_f W R_\beta(\mathbf{W})$:

$$\mathbf{W} + \beta\mathbf{W} - \beta\mathbf{W}\mathbf{W}^T\mathbf{W} = (1+\beta)\mathbf{W} - \beta\mathbf{W}\mathbf{W}^T\mathbf{W}; \tag{4.38}$$

To avoid high computational costs, this rule is applied only once in every iteration of gradient descent. After **W** is acquired, Conneau et al. (2017) use Procruste's method as refinement, to build a "high-quality dictionary". Word pairs, whose translations are *mutually* nearest neighbors, are taken as an immutable basis, around which non-mutual translations are re-arranged. A more detailed description of Procruste's method can be found in Section 4.2.2.

The corresponding translation for a word vector is then among its nearest neighbors after its multiplication with **W**. For a precise identification, a technique called *cross-domain similarity local scaling* (from now on, csls) is used, besides the traditional nearest neighbor search. Thereby, the similarity between a word in the source and the target language is calculated by

$$csls(\mathbf{W}\mathbf{x}_s, \mathbf{y}_t) = 2\cos(\mathbf{W}\mathbf{x}_s, \mathbf{y}_t) - r_T(\mathbf{W}\mathbf{x}_s) - r_S(\mathbf{y}_t) \tag{4.39}$$

where $\mathbf{x}_s$ is any embedding vector in the source language, $\mathbf{y}_t$ one of the k nearest neighbors of $\mathbf{W}\mathbf{x}_s$, and

$$r_T(\mathbf{W}\mathbf{x}_s) = \frac{1}{k} \sum_{\mathbf{y}_t \in \textit{nearest-neighbour}_k(\mathbf{W}\mathbf{x}_i)} \cos(\mathbf{W}\mathbf{x}_s, \mathbf{y}_t) \tag{4.40}$$

and similarly

$$r_S(\mathbf{y}_t) = \frac{1}{k} \sum_{\mathbf{x}_s \in \textit{nearest-neighbour}_k(\mathbf{W}^T\mathbf{y}_i)} \cos(\mathbf{W}^T\mathbf{y}_t, \mathbf{x}_s) \tag{4.41}$$

denote the average cosine value of the angle to their k nearest neighbours. In areas with a low occurrence of word vectors, this approach strengthens the similarity between source and translation, while in densly populated regions, the similarity is weakened. Doing so mitigates the so-called hubness-problem, where in areas with many data points, nearest neighbours can confer misleading information. The authors test CSLS against nearest-neighbour as standard and an alternative called inverted soft-max (ISF) (Smith et al., 2017), which selects the best translation word by looking for the target word that has the highest probability to translate back to the original source word.
Figure 4.2 pictures a sketch on the overall translation procedure:

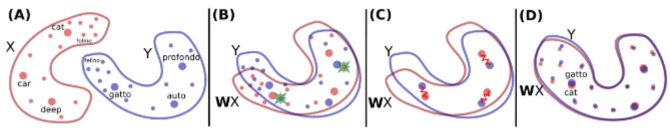


FIGURE 4.2: Overview over the Approach

Subplot (A) shows visualizations of English (X) and Italian (Y) word embeddings, represented by red and blue dots. The larger the dot, the more frequent the corresponding word is. (B) With the adversarial approach, matrix **W** is determined, which rotates the word vectors such that source and target space are roughly aligned. Green stars denote those words being randomly fed to the discriminator. (C) The mapping **W** is improved using Procruste's method around anchor points found in (B). (D) In the last step, words can be translated used *csls*. The space between vectors in dense areas is stretched, whereas in sparse regions, the distances are shrinked (for instance, around the words *gatto-cat*).

4.2 Analytical Approaches

From an analytical perspective, the task is to overlay two sets of high-dimensional data points optimally. There are two ways to proceed: Either generally with SGD, or by using a closed form solution with certain limitations.

4.2.1 Neural Network Optimization

This approach is described in (Mikolov et al., 2013). The authors observe a similar pattern in the distribution of word vectors as (Conneau et al., 2017) in Figure 4.2:

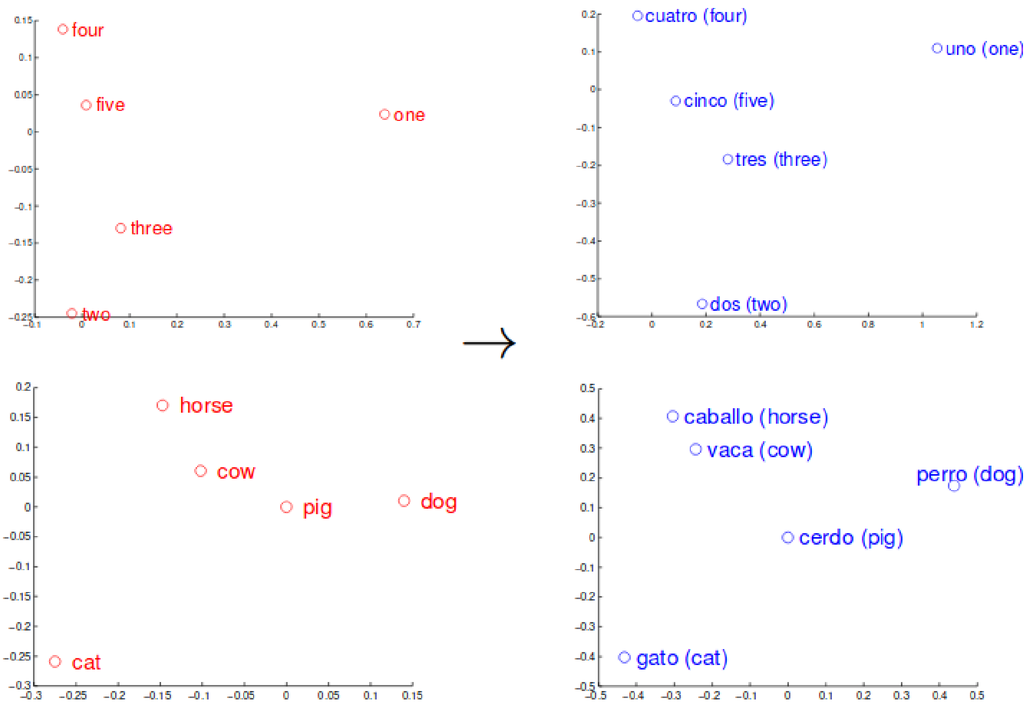


FIGURE 4.3: Observation by Mikolov et al. (2013): Similar Structure in the Distribution of Word Vectors in English and Spanish

Plots in Figure 4.3 shows that the distribution of word vectors of common number and animal terms in English and Spanish, projected into a two-dimensional space with principal component analysis and manually rotated afterwards, coincide in their shape. This motivates a semi-supervised translation process, in which preselected seed dictionary entries function as anchor points, around which the word vector distributions are rotated to minimize the distance between both data clouds. More formally, the approach starts with a set of n aligned word-embedding pairs $\{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1, \ldots, n\}$, with $\mathbf{x}_i \in \mathbb{R}^{d_S}$ and $\mathbf{y}_i \in \mathbb{R}^{d_T}$ being the embedding representations in the source respectively target language. Word vectors are trained with CBOW and Skip-Gram (see Section 3.2.2.1). The translation matrix $\mathbf{W} \in \mathbb{R}^{d_T \times d_S}$ maps the predefined seed pairs $(\mathbf{x}_i, \mathbf{y}_i)$ onto each other by projecting word vectors from the d_S-dimensional source embedding space into the d_T-dimensional target embedding space. To find the optimal $\mathbf{W}$, the objection function

$$\min_{\mathbf{W}} \sum_{i=1}^{n} \|\mathbf{W}\mathbf{x}_i - \mathbf{y}_i\|^2 \tag{4.42}$$

searches for any matrix $\mathbf{W}$ that minimizes the squared distance between the mappings of all $\mathbf{x}_i$ in $\mathbb{R}^{d_T}$ and their translations, $\mathbf{y}_i$, with SGD operating on a one-level NN. Figuratively speaking, the objective function above shifts and rotates all source word embedding pairs, until the positions of the anchor words correspond with their target words. In further experiments, Mikolov et al. (2013) enhance $\mathbf{W}$ with a weighted combination between the results of translation matrix and edit distance,

which can be useful for related languages (for example, *emotions* in English and *emociones* in Spanish share a long substring).
Since the mappings do not exactly correspond with data points in the target embedding space, the closest word vector $\mathbf{y}_j$ according to the cosine distance is again employed as translation for any vector $\mathbf{x}_i$ in the source space:

$$\mathbf{y}_j = \arg\max_{\mathbf{y}'} \cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}') \tag{4.43}$$

Certain cosine-thresholds are additionally set to improve translation quality; doing so ensures that not any nearest neighbor is taken as translation, and ought to improve translation precision.

4.2.2 Procruste's Problem

Another way of finding a translation matrix $\mathbf{W}$ leads to solving *Procruste's problem* (cf. Artetxe et al. (2016), Artetxe et al. (2017), and Schönemann (1966)). The problem setting remains the same as last section; two data clouds need to be aligned by a linear operation. In contrast to the SGD approach seen in the last chapter, a closed form for equation (4.42) can be derived, if some restrictions are met. Exemplarily, (Artetxe et al., 2017) is presented here, for their thorough investigation of the approach.
Starting point are CBOW word vectors trained with negative sampling (Section 3.2.2.1). A seed dictionary is formalized through an adjacency matrix $\mathbf{D} \in \{0,1\}^{d_S \times d_T}$, with d_S being the source and d_T the target embedding dimension:

$$\mathbf{D}[i][j] = \begin{cases} 1, \text{ if the } i\text{th source and } j\text{th target word are aligned,} \\ 0 \text{ otherwise.} \end{cases} \tag{4.44}$$

Beginning with only a seed dictionary, $\mathbf{D}$ is successively updated to store induced translations for the remaining vocabulary. The objective to determine the actual translation matrix $\mathbf{W}$ is (similar to Mikolov et al. (2013))

$$\mathbf{W}^\star = \arg\min_{\mathbf{W}} \sum_i \sum_j \mathbf{D}[i][j] \left\| \mathbf{x}_i\mathbf{W} - \mathbf{y}_j \right\|^2 \tag{4.45}$$

where $\mathbf{x}_i$ and $\mathbf{y}_j$ are word embeddings of the source and the target language. It aims to find that $\mathbf{W}$, which minimizes the distance between previously aligned word vectors.

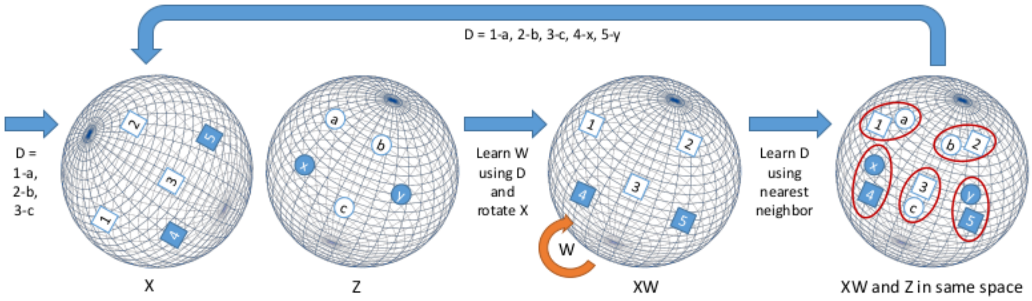


FIGURE 4.4: Visualization of the Process

In order to apply the closed form solution later on, two constraints have to be fulfilled: First, **W** needs to be orthogonal, which preserves monolingual invariance and improves translations, as also noted in section 4.1.2.. Secondly, the number of rows and columns has to correspond. In the following, **W** is assumed to be orthogonal.
Let matrices **X** and **Y** store the source and target vectors $\mathbf{x}_i$ and $\mathbf{y}_j$ as rows. Length-normalized, the embedding vectors in **X** and **Y** can then be compared by simply taking the dot-product, as opposed to cosine-similarity. Additionally, Artetxe et al. (2016) prove

$$\begin{aligned}\arg\min_{\mathbf{W}} \sum_i \sum_j \mathbf{D}[i][j] \left\| \frac{\mathbf{X}[i][:]}{\|\mathbf{X}[i][:]\|}\mathbf{W} - \frac{\mathbf{Y}[j][:]}{\|\mathbf{Y}[j][:]\|} \right\|^2 = \\ \arg\max_{\mathbf{W}} \sum_i \sum_j \mathbf{D}[i][j] cos(\mathbf{X}[i][:]\ \mathbf{W}, \mathbf{Y}[i][:]),\end{aligned} \tag{4.46}$$

i.e. minimizing the distance between length-normalized projected source and target embedding vectors maximizes the cosine similarity.
Furthermore, all vectors are mean centered, as in (Artetxe et al., 2016). The authors elaborate that

$$\begin{aligned}\arg\min_{\mathbf{W}} \sum_i \sum_j \mathbf{D}[i][j]\ \|\mathbf{X}[i][:]\ \mathbf{W} - \mathbf{Y}[j][:]\|^2 = \\ \arg\max_{\mathbf{W}} \sum_i \sum_j \mathbf{D}[i][j]\ cov(\mathbf{X}[i][:]\ \mathbf{W}, \mathbf{Y}[i][:]),\end{aligned} \tag{4.47}$$

hence minimization the distance between mean-centered projected source and target embedding vectors means maximizing the covariance between the bilingual seed pairs. Besides, dimension-wise mean-centering also results in an almost-zero dot product of two randomly chosen word vectors. That is, because every vector is centered around its own mean, randomly selected words are likely to be unrelated, and are almost orthogonal to each other.
In the closed form solution for Procruste's Problem given in the appendix of (Artetxe et al., 2016), embedding matrices are already aligned, meaning, the ith row vectors in **X** and **Y** correspond according to their seed dictionary. Thus, the solution needs to be slightly adjusted: Fortunately, **D** can also be viewed as an incomplete *permutation*

matrix with null rows[1], interchanging the rows of $\mathbf{Y}$. Therefore, $\mathbf{Y}$ can be multiplied by $\mathbf{D}$ to restore correspondences between the row word vectors in $\mathbf{X}$ and $\mathbf{Y}$. With this modification, the solution of Artetxe et al. (2016) unfolds to

$$\begin{aligned}
\mathbf{W}^{\star} = & \arg\min_{\mathbf{W}} \sum_i \sum_j \mathbf{D}[i][j] \left\| \mathbf{x}_i \mathbf{W} - \mathbf{y}_j \right\|^2 \Leftrightarrow \\
& \arg\min_{\mathbf{W}} \sum_{\substack{i \\ \text{s.t. } (\mathbf{x}_i, \mathbf{y}_i) = (s,t)}} \| \mathbf{x}_i \mathbf{W} - (\mathbf{DY})[i][:] \|^2 \Leftrightarrow \\
& \arg\min_{\mathbf{W}} \sum_{\substack{i \\ \text{s.t. } (\mathbf{x}_i, \mathbf{y}_i) = (s,t)}} (\| \mathbf{x}_i \mathbf{W} \|^2 + \| (\mathbf{DY})[i][:] \|^2 - 2\mathbf{x}_i \mathbf{W} (\mathbf{Y}^T \mathbf{D}^T)[i][:]) \Leftrightarrow \\
& \arg\max_{\mathbf{W}} Tr(\mathbf{X} \mathbf{W} \mathbf{Y}^T \mathbf{D}^T) \Leftrightarrow \\
& \arg\max_{\mathbf{W}} Tr(\mathbf{W} \mathbf{Y}^T \mathbf{D}^T \mathbf{X}) \Leftrightarrow \\
& \arg\max_{\mathbf{W}} Tr(\mathbf{X} \mathbf{W} \mathbf{Y}^T \mathbf{D}^T)
\end{aligned} \tag{4.48}$$

The fourth line results from the fact that

$$\begin{aligned}
\| \mathbf{x}_i \mathbf{W} \|^2 &= (\mathbf{x}_i \mathbf{W})(\mathbf{x}_i \mathbf{W})^T \\
&= \mathbf{x}_i \mathbf{W} \mathbf{W}^T \mathbf{x}_i^T \\
&= \mathbf{x}_i \mathbf{x}_i^T,
\end{aligned} \tag{4.49}$$

since $\mathbf{W}$ is orthogonal, and therefore $\mathbf{W}\mathbf{W}^T = \mathbf{I}$. Together with $\| (\mathbf{DY})[i][:] \|^2$, it can be ignored, as it does not affect the solution for any $\mathbf{W}$. This leaves

$$\arg\min_{\mathbf{W}} \sum_i -2\mathbf{x}_i \mathbf{W} (\mathbf{Y}^T \mathbf{D}^T)[i][:] \tag{4.50}$$

which is equivalent to

$$\arg\max_{\mathbf{W}} \sum_i \mathbf{x}_i \mathbf{W} (\mathbf{Y}^T \mathbf{D}^T)[i][:] \tag{4.51}$$

Again, factor two can be dropped, as it does not affect the solution. In the next line, the trace-operator $Tr(\cdot)$ in introduced (see (Schönemann, 1966) for details), which returns the sum of all elements on the main diagonal of a matrix. Lastly, because of the *cyclic property* of the trace operator, the order of matrix multiplication can be rearranged (Artetxe et al., 2016).

To solve the maximization problem, Schönemann (1966) and Artetxe et al. (2017)) take the SVD of $\mathbf{X}^T \mathbf{DY}$:

$$\begin{aligned}
SVD(\mathbf{X}^T \mathbf{DY}) &= \mathbf{U\Sigma V}^T \Leftrightarrow \\
SVD((\mathbf{X}^T \mathbf{DY})^T) &= (\mathbf{U\Sigma V}^T)^T \Leftrightarrow \\
SVD(\mathbf{Y}^T \mathbf{D}^T \mathbf{X}) &= \mathbf{V\Sigma U}^T
\end{aligned} \tag{4.52}$$

[1] Unless each source word has *at most* one translation; cf. http://encyclopediaofmath.org/index.php?title=Permutation_matrix&oldid=36223 [Accessed: 6.8.2020]

Note that transposing $\mathbf{\Sigma}$ in the bottom line yields again $\mathbf{\Sigma}$, because it is a diagonal matrix. The last line of (4.48) can hence be rewritten as

$$\begin{aligned}\arg\max_{\mathbf{W}} Tr(\mathbf{W}\mathbf{Y}^T\mathbf{D}^T\mathbf{X}) &= \arg\max_{\mathbf{W}} Tr(\mathbf{W}\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T) \\ &= \arg\max_{\mathbf{W}} Tr(\mathbf{U}^T\mathbf{W}\mathbf{V}\mathbf{\Sigma}) \\ &.\end{aligned} \tag{4.53}$$

$\mathbf{U}$, $\mathbf{V}^T$, and $\mathbf{W}$ are orthogonal matrices, and so their multiplication gives again an orthogonal matrix. The cyclic property is once more exploited to rearrange matrices within $Tr(\cdot)$. $\mathbf{\Sigma}$ contains the largest-possible entries on its diagonal, which means, that the trace is maximal, if the entries in $\mathbf{\Sigma}$ remain. This is the case, if $\mathbf{U}^T\mathbf{W}\mathbf{V} = \mathbf{I}$ holds true, so

$$\mathbf{W} \stackrel{!}{=} \mathbf{U}\mathbf{V}^T. \tag{4.54}$$

More loosely speaking, for maximizing the trace, first SVD is required, as it returns the maximal diagonal values. Secondly, to keep this solution maximally, $\mathbf{U}^T\mathbf{W}\mathbf{V}$ is supposed to return the identity matrix, which is adhered to equation (4.54).
With this solution at hand, Artetxe et al. (2017) use now an iterative process to update dictionary $\mathbf{D}$: Starting with an initial $\mathbf{D}$, the objective becomes maximized; then,

$$\mathbf{D}[i][j] = \begin{cases} 1, \text{ if } \mathbf{x}_i\mathbf{W}\mathbf{y}_j^T \text{ is maximal} \\ 0 \text{ otherwise.} \end{cases} \tag{4.55}$$

As noted at the beginning of the chapter, the embedding vectors are length-normalized, which is why, the dot-product poses a sufficient similarity measure. The process is iterated, until the average of all updates converge.

4.3 Graphical Approaches

Taking a graphical view, words are organized as nodes in a co-occurrence graph. Before the graphical approaches are laid out in detail, the necessary notation for this section is briefly explained.
A graph

$$G = (V, E, w) \tag{4.56}$$

consists of a set of vertices,

$$V = \{v_1, v_2, \ldots, v_n\}, \tag{4.57}$$

a set of edges,

$$E = \{(v_i, v_j) \mid v_i \text{ and } v_j \text{ are connected}\} \subseteq V \times V, \tag{4.58}$$

weighted by a function

$$w : V \times V \mapsto \mathbb{R} \tag{4.59}$$

which takes edges and returns a real-valued weight.
Each vertex v_i has a set of outgoing links,

$$O(v_i) = \{v_j \mid (v_i, v_j) \in E\}, \tag{4.60}$$

denoting all *adjacent* nodes to v_i, as well as a set of incoming links,

$$I(v_i) = \{v_j \mid (v_j, v_i) \in E\}, \tag{4.61}$$

i.e. all nodes that have links *pointing to* v_i. Every graph can be represented by a matrix $\mathbf{M} \in \mathbb{R}^{|V| \times |V|}$, with the entry $\mathbf{M}[i][j]$ corresponding to the weight associated with the edge (v_i, v_j).
All graphical methods presented here are based on the concept of random walks, specifically on graphs. A random walk simulates a motion on a graph, starting from some vertex, and wandering at each time step to an adjacent node (Lovász et al., 1993). The next node is usually drawn uniformly from the set of neighbours. Let $\widetilde{\mathbf{M}}$ henceforth be a row-normalized matrix representation of a graph, where all entries in a row sum up to one[2]. Thus, $\widetilde{\mathbf{M}}$ is often called *transition* or *stochastic matrix*, because each entry $\widetilde{\mathbf{M}}[i][j]$ gives the *transition probability* of going from vertex v_i to v_j, corresponding to the links of a Markov chain. The probability of reaching v_j after k steps starting from v_i is given by $\widetilde{\mathbf{M}}^k[i][j]$ (Erkan and Radev, 2004). The *stationary* distribution p of $\widetilde{\mathbf{M}}$ is defined by

$$\forall i \in \{1, \ldots, |V|\} : \lim_{k \to \infty} \widetilde{\mathbf{M}}^k[i][:] = \mathbf{p}. \tag{4.62}$$

It formalizes the intuition that after infinitely many steps, the starting point is no longer relevant for ending up at a certain vertex. Each node has then a fixed probability, with which the random walker is attracted to it. The existence of a stationary distribution is guaranteed by the Perron-Frobenius theorem (Pillai et al., 2005), if a square matrix has positive entries and is both irreducible an aperiodic. Without going into the mathematical details of irreducibility and aperiodicity, the intuition behind these concepts is as follows (Erkan and Radev, 2004): A transition matrix is said to be irreducible, if every vertex can be reached by every other vertex, i. e. it is impossible to arrive at a node which cannot be left. Aperiodicity means that the number of steps it takes to visit the same node again follows no regularity. If both criteria meet, $\widetilde{\mathbf{M}}^k$ converges for a large, but finite, k.
One algorithm in particular, which exploites the stationary distribution, is PAGERANK (Page et al., 1999). The subsequent graphical approaches, as well as the one presented in this thesis, are extensions of its model. The idea behind PAGERANK is now to model the behaviour of a random surfer on the web (Page et al., 1999). The linkage in this case is binary: Either, a website does link to another one, or it does

[2]In case all entries are ≥ 0, this can be done simply by division with the sum of all entries. Otherwise, the soft-max function can be utilized (cf. equation 3.4)

not. This corresponds to an edge weight of either one or zero. Starting at some website, the surfer randomly follows one outgoing link at each step. Without additional prior information, the initial website is chosen uniformly and all outgoing links are treated equally likely. Intuitively, a random surfer should be drawn to websites with higher PAGERANK scores (than to ones with lower scores). Technically, this idea is modeled by the following equation:

$$PR(v_i) = \sum_{v_j \in I(v_i)} \frac{1}{|O(v_j)|} PR(v_j) \tag{4.63}$$

In the model, the PAGERANK score of a page, i.e. a vertex v_i, depends on the PAGERANK score of its adjacent vertices, v_j, weighted by the inverse number of their outgoing links. A vertex with a low number of outgoing arcs contributes more to the score of its neighbors than a vertex with many outgoing arcs.

In order to capture the random element in the surfer's behaviour better, a damping factor $\mathtt{d} \in [0, 1]$ is introduced, which accounts for restarts within the random walk (see Brin and Page (1998) and Erkan and Radev (2004)):

$$PR(v_i) = \frac{\mathtt{d}}{|V|} + (1 - \mathtt{d}) \sum_{v_j \in I(v_i)} \frac{1}{|O(v_j)|} PR(v_j) \tag{4.64}$$

Hereby, the PAGERANK score is balanced between a uniformly chosen vertex $v \in V$ and the original PAGERANK calculation. Thus, $\mathtt{d}$ also *dampens* the influence of nodes which are far away from the starting point. This has also another advantage: Since it cannot be guaranteed that $\widetilde{\mathbf{M}}$ is irreducible and aperiodic, $\mathtt{d} < 1$ forces the values to converge. Brin and Page empirically determine a value of $\mathtt{d} = 0.15$ to work best in most settings. Another description interchanges $\mathtt{d}$ and $1 - \mathtt{d}$, leading to

$$PR(v_i) = \frac{(1 - \mathtt{d})}{|V|} + \mathtt{d} \sum_{v_j \in I(v_i)} \frac{1}{|O(v_j)|} PR(v_j) \tag{4.65}$$

It can be easily seen that with an appropriately adjusted $\mathtt{d}$, both representations are equivalent.

For convenience, the recursive instruction can be rewritten in matrix notation.

$$\mathbf{p} = \left(\mathtt{d}\mathbf{U} + (1 - \mathtt{d})\widetilde{\mathbf{M}}\right)^T \mathbf{p} \tag{4.66}$$

$\mathbf{p}$ is again the vector converging to the stationary distribution, $\mathbf{U}$ is a matrix with all values being set to $\frac{1}{|V|}$.

Besides its application in search engines, PAGERANK has also been employed in other tasks, such as text summarization (Mihalcea, 2004) and word sense disambiguation (Agirre and Soroa, 2009).

In the upcoming sections, the idea of PAGERANK is extended to multiple graphs. Instead of calculating the score for only one vertex, it is calculated for two vertices in different graphs, indicating the degree of similarity between them. This also changes the resulting data structure, from a vector containing the PAGERANK scores for every vertex, to a matrix containing the pairwise similarities for each vertex pair.

4.3.1 SIMRANK

SIMRANK (Jeh and Widom, 2002) operates like PAGERANK on directed, unweighted graphs. The similarity between two nodes v_i and v_j in a graph with adjacency matrix **M** is defined by

$$s(v_i, v_j) = \frac{\mathsf{c}}{|I(v_i)||I(v_j)|} \sum_{v_k \in I(v_i)} \sum_{v_l \in I(v_j)} s(v_k, v_l) \tag{4.67}$$

Originally developed for determining the similarity between nodes within the same graph, $s(v_i, v_j) = 1$ if $v_i = v_j$. If $v_i \neq v_j$, $s(v_i, v_j)$ depends recursively on the sum of the similarities between all pairs of nodes with links pointing to v_i and v_j, normalized by the potential maximum degree of association, $|I(v_i)||I(v_j)|$. $\mathsf{c} \in [0, 1]$ gives, as the authors note, "the rate of decay", which models the decreasing importance of affinities between distant pairs of vertices. The similarity between all pairs is then stored in a matrix, **S** (called **R** in (Jeh and Widom, 2002)). In case of isolated nodes, which do not have any incoming edges, the similarity is set to zero by default, to avoid division by zero.

Dorow et al. (2009) now use SIMRANK to determine similarities between nodes of otherwise unrelated graphs, which model word-to-word relations. Therefore, they first rewrite (4.67) in matrix notation to facilitate computation,

$$\begin{aligned}
s(v_i, v_j) &= \frac{\mathsf{c}}{|I(v_i)||I(v_j)|} \sum_{v_k \in I(v_i)} \sum_{v_l \in I(v_j)} s(v_k, v_l) \qquad (4.68)\\
&= \frac{\mathsf{c}}{|O(v_i)||O(v_j)|} \sum_{\substack{v_k \in O(v_i) \\ v_l \in O(j)}} \mathbf{S}[k][l] \\
&= \frac{\mathsf{c}}{|O(v_i)||O(v_j)|} \sum_{\substack{v_k \in O(v_i) \\ v_l \in O(j)}} \mathbf{M}[i][k]\ \mathbf{M}[j][l]\ \mathbf{S}[k][l] \\
&= \mathsf{c} \sum_{\substack{v_k \in O(v_i) \\ v_l \in O(j)}} \frac{\mathbf{M}[i][k]}{|O(v_i)|} \frac{\mathbf{M}[j][l]}{|O(v_j)|} \mathbf{S}[k][l] \\
&= \mathsf{c} \sum_{\substack{v_k \in O(v_i) \\ v_l \in O(j)}} \frac{\mathbf{M}[i][k]}{\sum_v \mathbf{M}[i][v]} \frac{\mathbf{M}[j][l]}{\sum_v \mathbf{M}[j][v]} \mathbf{S}[k][l] \\
&= \mathsf{c} \sum_{\substack{v_k \in O(v_i) \\ v_l \in O(j)}} \widetilde{\mathbf{M}}[i][k]\ \widetilde{\mathbf{M}}[j][l]\ \mathbf{S}[k][l] \\
&= \mathsf{c} \cdot \left(\sum_{\substack{v_k \in O(v_i) \\ v_l \in O(j)}} \widetilde{\mathbf{M}}[i][k]\ \mathbf{S}[k][l]\ \widetilde{\mathbf{M}}^T[j][l] \right)
\end{aligned}$$

$$= \mathsf{c} \cdot \left(\widetilde{\mathbf{M}}\mathbf{S}\widetilde{\mathbf{M}}^T\right)[i][j]$$

where $\widetilde{\mathbf{M}}$ denotes again the row-normalized adjacency matrix $\mathbf{M}$. This gives for the kth iteration of the calculation:

$$\mathbf{S}^k = \mathsf{c}\widetilde{\mathbf{M}}\mathbf{S}^{k-1}\widetilde{\mathbf{M}}^T \tag{4.69}$$

It is worth noting that the set of adjacent nodes changes: While Jeh and Widom (2002) use the set of incoming nodes with arcs pointing *to* the vertices in question, Dorow et al. (2009) make use of the set of nodes to which the vertices in question are pointing to. Though, the overall behavior of the similarity measure remains unaffected by this modification. Dorow et al. (2009) further extend the approach to *weighted typed* connections in the graph. Doing so allows for a more fine-grained analysis, since both the strength of contextual co-occurrences and linguistic information, such as the part-of-speech tags of the word and context term, are considered. As long as the adjacency matrix is row-normalized, weights can be chosen arbitrarily. By introducing typed edges $t \in \mathcal{T}$, (4.69) becomes

$$\mathbf{S}^k = \frac{\mathsf{c}}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\widetilde{\mathbf{M}}_t\mathbf{S}^{k-1}\widetilde{\mathbf{M}}^T \tag{4.70}$$

In order to use SIMRANK on two graphs, $G = (V, E)$ and $G' = (V', E')$, with row-normalized adjacency matrices $\widetilde{\mathbf{M}}, \widetilde{\mathbf{M}'}$, one needs a (sub)set of node pairs from both graphs with predefined similarities (Dorow et al., 2009). If a complete probability distribution over all pairs of vertices is desired, the sum of all initial correspondences in this preliminary similarity matrix $\mathbf{S}^0$ ought to yield one (cf. PAGERANK). In case of translation, there are two options for instantiation: Supervised, which means that for some terms the corresponding translations need to be manually assigned, and unsupervised, where all correspondences are instantiated with the same value, preferably $\frac{1}{|V|\cdot|V'|}$.

The formula for this modified version of SIMRANK follows the reasoning of (4.69)

$$\mathbf{S}^k = \mathsf{c}\widetilde{\mathbf{M}}\mathbf{S}^{k-1}\widetilde{\mathbf{M}'}^T \tag{4.71}$$

And analogously, for typed edges:

$$\mathbf{S}^k = \frac{\mathsf{c}}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\widetilde{\mathbf{M}}_t\mathbf{S}^{k-1}\widetilde{\mathbf{M}'}^T \tag{4.72}$$

In analogy to PAGERANK's random surfer on the web, one can picture the algorithm as a traveler jumping from one tuple of vertices $(v_i, v'_j) \in V \times V'$ to another, preferring transitions with high relative similarities. As a result, instead of a probability vector, a matrix is obtained, whose entries sum up to one. The desired outcome is depicted in the following figure, where a similar English and German graph are aligned, such that the corresponding translations are correctly identified:

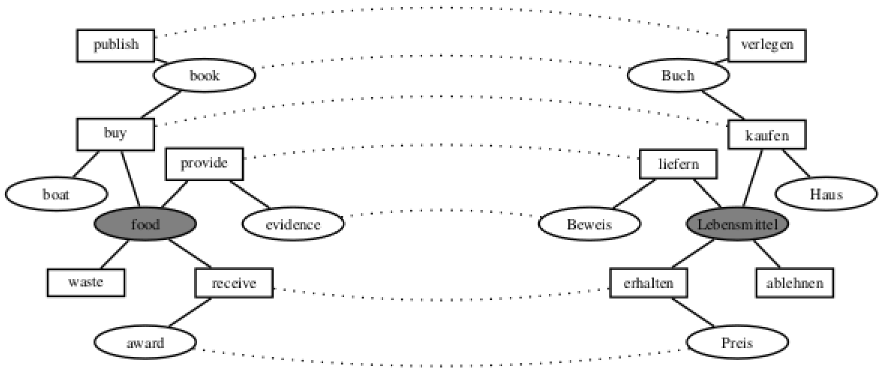


FIGURE 4.5: Correctly aligned English and German Network

Once having obtained similarity matrix **S**, Dorow et al. (2009) translate words through selecting the largest value for each (target word) row, and matching its corresponding (goal word) column:

$$w_x = \arg\max_x \mathbf{S}[i][x]. \tag{4.73}$$

Concerning its run-time, SIMRANK takes about $\mathcal{O}(n^3)$ calculatory steps for matrix multiplication, and a vicinity of $\mathcal{O}(n^2)$ in memory, for a total amount of n words (Rothe and Schütze, 2014).

4.3.2 COSIMRANK

A variant of SIMRANK is COSIMRANK (Rothe and Schütze, 2014). As for SIMRANK, starting point is a directed, unweighted graph $G = (V, E)$ and its adjacency matrix **M**, where nodes represent words from a vocabulary. The goal is again to measure similarity between nodes, at first in one graph, and then extending the methodology to two networks. In contrast to SIMRANK, the similarity is not recursively calculated between nodes, but between personalized PAGERANK (hence, PPR) vectors (Haveliwala, 2002) of two vertices. Following Haveliwala (2002), the PPR vector for node $v_i \in V$ is defined by

$$\mathbf{p}^k(v_i) = \mathtt{d}\widetilde{\mathbf{M}}\mathbf{p}^{k-1}(v_i) + (1-d)\mathbf{p}^0(v_i), \tag{4.74}$$

where $\widetilde{\mathbf{M}}$ is again a row-normalized Markov transition matrix and $\mathbf{p}^0$(i) the ith canonical vector, i.e. vector of size $|V|$ with $\mathbf{p}[i] = 1$ and $\mathbf{p}[j] = 0\ \forall j \neq i$. The resulting vector, $\mathbf{p}^k(i)$, depends on the node from which the random surfer began his journey. Instead of any random node, the surfer always restarts on the same vertex with probability $1 - \mathtt{d}$. This way, every vertex is assigned to an individual, therefore *personal*, stationary distribution. Rothe and Schütze (2014) simplify above expression by setting $\mathtt{d} = 1$, which allows a compact matrix notation later on:

$$\mathbf{p}^k(v_i) = \widetilde{\mathbf{M}}\mathbf{p}^{k-1}(v_i) \tag{4.75}$$

The decay factor is re-introduced during similarity computation, which is carried out by the dot-product of the PPR vectors, comparable to cosine similarity (cf. equation (3.3)). Since the vectors form probability distributions, they are already of finite length, and thus do not need to be normalized. In mathematical terms, row-normalizing means dividing the vector by its ℓ_1-norm, whereas in the cosine similarity, the vector is normalized by its ℓ_2-norm. This minor modification becomes later relevant, when assessing convergence.
Instead of taking only the fully-converged distributions, Rothe and Schütze (2014) calculate a stepwise comparison between both vertices' PPR after each iteration. The benefit of doing so is that vertices, which are distant in G, receive nonetheless a none-zero weight in PPR vectors, thereby diluting the result.
The similarity measure between vertices $v_i, v_j \in V$ is given by:

$$s(v_i, v_j) = \sum_{k=0}^{\infty} \mathsf{c}^k \langle \mathbf{p}^k(i), \mathbf{p}^k(j) \rangle \tag{4.76}$$

c is the newly added decay factor, and $\langle \cdot, \cdot \rangle$ denotes the aforementioned dot-product. A recursive equivalent of (4.76) would be

$$s^k(v_i, v_j) = \mathsf{c}^k \langle \mathbf{p}^k(v_i), \mathbf{p}^k(v_j) \rangle + s^{k-1}(v_i, v_j) \tag{4.77}$$

In matrix notation, this becomes

$$\begin{aligned}
\mathbf{S}^0 &= \mathbf{I} \\
\mathbf{S}^1 &= \mathsf{c}\widetilde{\mathbf{M}}\widetilde{\mathbf{M}}^T + \mathbf{S}^0 \\
\mathbf{S}^2 &= \mathsf{c}^2\widetilde{\mathbf{M}}^2 \left(\widetilde{\mathbf{M}}^T\right)^2 + \mathbf{S}^1 \\
&\vdots \\
\mathbf{S}^k &= \mathsf{c}^k\widetilde{\mathbf{M}}^k \left(\widetilde{\mathbf{M}}^T\right)^k + \mathbf{S}^{k-1},
\end{aligned} \tag{4.78}$$

with the matrix multiplication of $\widetilde{\mathbf{M}}^k \left(\widetilde{\mathbf{M}}^T\right)^k$ being the equivalent to the kth dot product of $\langle \mathbf{p}^k(v_i), \mathbf{p}^k(v_j) \rangle$. As Rothe and Schütze (2014) prove, (4.78) can be rewritten as

$$\mathbf{S}^k = \mathsf{c}\widetilde{\mathbf{M}}\mathbf{S}^{k-1}\widetilde{\mathbf{M}}^T + \mathbf{S}^0, \tag{4.79}$$

which emphasizes the resemblance with SIMRANK (recall formula (4.69)).With typed edges $t \in \mathcal{T}$, (4.79) is adjusted such that

$$\mathbf{S}^k = \left(\frac{\mathsf{c}}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \widetilde{\mathbf{M}}_t \mathbf{S}^{k-1} \widetilde{\mathbf{M}}^T \right) + \mathbf{S}^0. \tag{4.80}$$

$\widetilde{\mathbf{M}}_t$ describes the adjacency matrix for the specific edge type t. In both set-ups, the closest term w_x for any word w_i is then the one with the largest row-vector entry.

Moving on to node similarities between graphs, the second graph is denoted by $G' = (V', E')$, with $\mathbf{M'}$ as adjacency matrix. Equation (4.76) can then be extended to:

$$s(v_i, v'_j) = \sum_{k=0}^{\infty} \sum_{(n,m)\in\mathbf{S}^0} \mathbf{p}_u^k(v_i)[n]\ \mathbf{q}(v'_j)[m] \tag{4.81}$$

where $\mathbf{p}^k(v_i)[n]$ is the nth entry of the PPR vector for $v_i \in V$ after the kth iteration, and $\mathbf{q}^k(v'_j)[m]$ likewise the mth entry of the PPR vector for $v'_j \in V'$ for the kth step. Reformulated with matrices, this yields

$$\mathbf{S}^k = \mathsf{c}^k \widetilde{\mathbf{M}}^k \mathbf{S}^0 \left(\mathbf{M'}^T\right)^k + \mathbf{S}^{k-1} \tag{4.82}$$

Following Dorow et al. (2009), $\mathbf{S}^0 \in \mathbb{R}^{|V|\times|V'|}$ contains the initial seed dictionary. For reasons of space complexity, in the typed version, only the last traveled edge is required to be of the same type:

$$\mathbf{S}^k = \frac{c}{|T|} \sum_{t\in T} \widetilde{\mathbf{M}}_t \mathbf{S}^{k-1} (\mathbf{M'}_t)^T + \mathbf{S}^0 \tag{4.83}$$

As in the case of SIMRANK, $\mathbf{S}$ can be perceived as change of basis between source and goal vector space. Analogously to determining the closest term within the same graph, translating a word is a matter of taking the largest value of the row-vector in question, and using the matching index as goal word.
For all variants of COSIMRANK, Rothe and Schütze (2014) guarantee convergence if $\mathsf{c} < 1$. Let $\mathbf{u}, \mathbf{v}$ be two ℓ_1 normalized vectors. Then, by Cauchy-Schwarz-inequality, it holds true that

$$\langle \mathbf{u}, \mathbf{v} \rangle \leq \|\mathbf{u}\|_2 \|\mathbf{v}\|_2 \tag{4.84}$$

and, since ℓ_1 is largest of all p-norms,

$$\|\mathbf{u}\|_2 \|\mathbf{v}\|_2 \leq \|\mathbf{u}\|_1 \|\mathbf{v}\|_1, \tag{4.85}$$

which gives

$$\langle \mathbf{u}, \mathbf{v} \rangle \leq 1. \tag{4.86}$$

Therefore, every entry $\mathbf{S}^k[i][j]$ is bounded by the geometric series, which converges for $\mathsf{c} < 1$:

$$\mathbf{S}^k[i][j] \leq \sum_{k=0}^{\infty} \mathsf{c} \cdot 1 = \frac{1}{1-\mathsf{c}} \tag{4.87}$$

Regarding runtime and memory consumption, COSIMRANK takes similar to SIMRANK $\mathcal{O}(n^3)$ calculation steps and $\mathcal{O}(n^2)$ memory, with n being the number of words in the vocabulary. For a smaller subsets of nodes, with a presumed lower average degree, the runtime can be further levelled down with sparse graph representations. Interested readers are referred to the original publication.

4.4 Proposed Method

The method proposed in this thesis is also a graphical one. Following the nomenclature of other $*$RANK approaches, it is henceforth called TRANSRANK. TRANSRANK aims to combine the advantages and mitigate the disadvantages of SIMRANK and COSIMRANK. Recalling the formula for SIMRANK,

$$s(v_i, v'_j) = \frac{c}{|O(v_i)||O(v'_j)|} \sum_{t \in T} \frac{1}{\mid O_t(v_i) \mid\mid O_t(v'_j) \mid} \sum_{v_k \in O_t(v_i)} \sum_{v'_l \in O_t(v'_j)} w_{ik} w'_{jl} s(v_k, v'_l) \tag{4.88}$$

it becomes clear that it takes only positive association into account. If two edges w_{ik}, w'_{jl} have a low transition probability, it does not contribute to the overall degree of similarity. This is, however, favourable. Knowing that a pair of words in different languages does *not* often co-occur with another pair of possibly similar words, adds more information to the process of translation. Furthermore, both do not make use of dense vector representations, but simple word-to-word relations.

Instead, TRANSRANK compares the similarity of two entries of input embedding matrices, $\mathbf{M}$ and $\mathbf{M'}$, not necessarily being row-normalized, with a similarity measure $sim(\cdot, \cdot)$. It uses an uninformed initial translation matrix $\mathbf{S} \in \mathbb{R}^{|V| \times |V'|}$. The formula for an entry $\mathbf{S}[i][j]$ is given by:

$$s(v_i, v'_j) = \frac{1 - \mathtt{d}}{|V||V'|} + \mathtt{d} \sum_{v_k \in V} \sum_{v'_l \in V'} \frac{sim(\mathbf{M}[k][i], \mathbf{M'}[l][j])}{\sum\limits_{v_{i'} \in V} \sum\limits_{v'_{j'} \in V'} sim(\mathbf{M}[k][i'], \mathbf{M'}[l][j'])} s(v_k, v'_l) \tag{4.89}$$

For each pair of nodes v_i, v'_j, in two graphs $G = (V, E)$ and $G' = (V', E')$, $s(v_i, v'_j)$ takes the similarity of all its adjacent pairs $s(v_k, v'_l)$, multiplied with the similarity between their edge weights $sim(\mathbf{M}[k][i], \mathbf{M'}[l][j])$ and normalized by the sum of similarities between all *outgoing* edge weights from v_k and v'_l,

$$\sum_{v_{i'} \in V} \sum_{v'_{j'} \in V'} sim(\mathbf{M}[k][i'], \mathbf{M'}[l][j']) \tag{4.90}$$

For the quotient to represent a useful probability distribution, $sim(\cdot, \cdot)$ has to fulfill the following criteria: Obviously, it has to be symmetric; the order, in which the matrix entries are plugged in, should not matter. If it would, $sim(v_i, v'_j) \neq sim(v'_j, v_i)$. Also, intuitively, small differences between the entries are supposed to result in high similarities, and large differences in small similarities (in mathematic terms, $sim(\cdot, cdot)$ should be *monotonically decreasing* with regard to the distance between the entries). Finally, the outcome ought to be bounded desirably in $(0, 1]$. Otherwise, one would risk an infinite number in the enumerator, preventing the sum from being one. These considerations lead to

$$sim(x, y) = \frac{1}{1 + \mathtt{c} \, \|x - y\|_2} = \frac{1}{1 + \mathtt{c}\sqrt{(x-y)^2}} \tag{4.91}$$

which satisfies the constraints stated above: Symmetry, decreasing monotonicity, and boundedness. Constant $\mathtt{c}$ additionally controls the influence of distance between both entries.

As can be seen from the plot, the similarity measure has a decreasing exponential development. This is a neat side effect, because high affinities are favored, whereas low similarities are punished. As a result, the result ought to become more robust; s_w and s_e return high weights only in unambiguous cases.

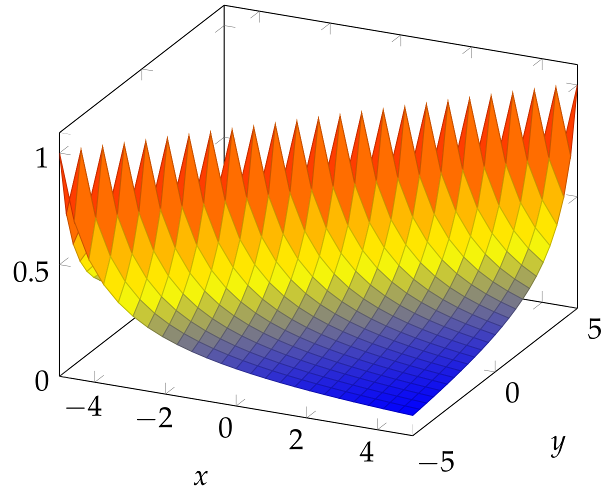


FIGURE 4.6: Plot of $sim(x, y, 1)$

Important for the consecutive application on word-embedding graphs is a modification for bipartite graphs proposed by Jeh and Widom (2002). In a bipartite graph $G_B = (V_B, E_B)$, the set of vertices V_B is categorized into two subsets $V_B = V_w \cup V_e$, with $V_w \cap V_e = \emptyset$ and $E_B = \{(u, v) | u \in V_w \wedge v \in V_e\}$. V_w can be perceived as word-, and V_e as embedding nodes; edges map from words to embeddings. Hence, two similarity measures are necessary, one for each set of nodes V_w, V_e:

$$s_w(v_i, v_j) = \frac{\mathtt{c}_w}{|O(v_i)||O(v_j)|} \sum_{v_k \in O(v_i)} \sum_{v_l \in O(v_j)} s_e(v_k, v_l) \tag{4.92}$$

$s_w(\cdot, \cdot)$, which measures the similarity between vertices in V_w, is defined by the normalized sum of all similarities between the nodes pointed to in V_e. Analogously, works

$$s_e(v_i, v_j) = \frac{\mathtt{c}_e}{|I(v_i)||I(v_j)|} \sum_{v_k \in I(v_i)} \sum_{v_l \in I(v_j)} s_w(v_k, v_l) \tag{4.93}$$

$\mathtt{c}_w$ and $\mathtt{c}_e$ are, again, (decaying) constants between zero and one. Following Dorow et al. (2009), this approach is now extended to two graphs.

As in SIMRANK's bipartite case, two similarity measures are necessary, one for each partition. Let therefore **M**, **M′** be the adjacency matrices for the bipartite graphs $G = (V_w \cup V_e, E)$, $G' = (V'_w \cup V'_e, E')$, where V_w, V'_w stand for the word (or, state) vertices (i.e., dimensions) V_e, V'_e denote the embedding vertices (dimensions). Then, the next two equations include these modifications:

$$s_w(v_i, v'_j) = \frac{1-\mathtt{d}}{|V||V'|} + \mathtt{d} \sum_{e_k \in V_e} \sum_{e'_l \in V'_e} \frac{sim(\mathbf{M}[i][k], \mathbf{M'}[j][l])}{\sum\limits_{v_{i'} \in V_w} \sum\limits_{v'_{j'} \in V'_w} sim(\mathbf{M}[i'][k], \mathbf{M'}[j'][l])} s_e(v_k, v'_l) \quad (4.94)$$

$$s_e(e_i, e'_j) = \frac{1-\mathtt{d}}{|V||V'|} + \mathtt{d} \sum_{v_k \in V_w} \sum_{v'_l \in V'_w} \frac{sim(\mathbf{M}[k][i], \mathbf{M'}[l][j])}{\sum\limits_{e_{i'} \in V_e} \sum\limits_{e'_{j'} \in V'_e} sim(\mathbf{M}[k][i'], \mathbf{M'}[l][j'])} s_w(v_k, v'_l) \quad (4.95)$$

e_i, e'_j denote the embedding dimensions i, j of G and G', respectively. The outcomes of s_w and s_e are stored in matrices $\mathbf{S}_w \in \mathbb{R}^{|V_w| \times |V'_w|}$ and $\mathbf{S}_e \in \mathbb{R}^{|V_e| \times |V'_e|}$, which can both be conceived as change of bases between word/ state and embedding vector spaces. Regarding the run-time, the approach takes for each entry in $\mathbf{S}_w$ $|V_w| \times |V'_w| \times |V_e| \times |V'_e|$, and therefore for the whole matrix $|V_w|^2 \times |V'_w|^2 \times |V_e| \times |V'_e|$ steps.
Fortunately, the outcomes of the nested sums in the denominator

$$\sum_{v_{i'} \in V_w} \sum_{v'_{j'} \in V'_w} sim(\mathbf{M}[i'][k], \mathbf{M'}[j'][l]) \quad (4.96)$$

can be stored in a separate tensor for all possible (k, l, i', j')-quadruples, which reduces the run-time to $|V_w| \times |V'_w| \times |V_e| \times |V'_e|$ steps. Likewise, the calculation of $\mathbf{S}_e$ needs also $|V_w| \times |V'_w| \times |V_e| \times |V'_e|$ steps. Under the assumption that embedding dimensions are bounded by the number of words in the vocabulary, n, and that vocabulary sizes are similar across the languages used, run-time and memory consumption (recalling the four-dimensional tensor) lie in the vicinity of $\mathcal{O}(n^4)$.
As can be seen for $\mathtt{d} \leq 1$, the series of calculated similarities decreases monotonically, and following the reasoning of PAGERANK, convergence can guaranteed.
For the translation process, only $\mathbf{S}_w$ is used, because its entries store the transition probabilities between one-hot encoded words or states, respectively. One could also translate the embedding vectors into each other, or interpolate the resulting goal vectors from one-hot and embedding vectors, such as Pennington et al. (2014). However, for starters, only $\mathbf{S}_w$ is employed.
Any word w_i in the source language is then translated by multiplying its associated one-hot vector $\mathbf{w}_i$ with $\mathbf{S}_w$ from the left-hand side:

$$\hat{\mathbf{w}}' = \mathbf{w}_i \mathbf{S}_w \quad (4.97)$$

$\hat{\mathbf{w}}'$ is the hypothesized translation of $\mathbf{w}_i$. Its actual corresponding term $\mathbf{w'}$ in the target language is determined by cosine similarity,

$$\mathbf{w}' = \max_{\mathbf{w}'_j \in \mathbf{M'}} cos(\hat{\mathbf{w}}', \mathbf{w}'_j) \quad (4.98)$$

meaning the closest row (i.e., word) vector $\mathbf{w}'_j$ in the target embedding matrix $\mathbf{M'}$. Translating a word from the target language into the source language works vice versa:

$$\mathbf{w} = \max_{\mathbf{w}_i \in \mathbf{M}} cos(\mathbf{w}_j' \mathbf{S}_w^T, \mathbf{w}_i) \tag{4.99}$$

Though it would be possible to translate source words into the target word with the largest row-value (see SIMRANK and COSIMRANK), the cosine similarity is utilized for the sake of a unified translation process. In case of state embeddings, a source word's vector comprises of the sum of all its state-embedding vectors. This vector is then analogously translated by $\mathbf{S}_w$, and compared to all state-composed word vectors in the target language. Taking just the largest values would not suffice, since their corresponding states might not form a valid word in the target vocabulary.
After all, for standard word-to-word translation, cosine-similarity and selecting the word with the largest row-entry, yields the same outcome.
In summary, this method is hypothesized to work well, especially with small datasets at hand, because of

Robust Exponential Similarity Function
Due to the its exponential behavior, $sim(\cdot, \cdot)$ assigns high similarity scores only if both parameters are close together.

Integration of Missing Context
If two words lack the same similar context, their association is reinforced in the same way, as if the very same context would be mutually shared.

The next chapter presents the experimental setup and the evaluation results.

Chapter 5

Experiments and Evaluation

In God we trust; all others must have data.
Cecil R. Reynolds (Reynolds, 1983)

This chapter describes how the proposed method and related approaches are evaluated. For a better readability, both the evaluation of word vectors and of the unsupervised dictionary induction is bundled here.
The first section presents the setup of the experiments. In the second part, results of the related work from the last chapter are shown and compared to the method advocated in this thesis.

5.1 Experimental Setup

This section covers the experiments around the proposed methods, starting with the underlying corpus and the extracted data set, continouing with an overview about the parameters, and ending with the evaluation procedure.

5.1.1 Corpus and Data Set

The choice of the corpus is crucial for an unbiased evaluation. Especially given a small data set, one cannot rely on any corpora, as the chance for collecting words from two unrelated domains, for instance politics and medicine, is high. In this case, translations between the languages would merely be analogies or metaphors of arbitrary distance. However, if the corpora are too close - parallel, at worst - the quality of TRANSRANK is no longer tested under genuine conditions.
These considerations lead to the `Wortschatz` corpora provided by the University of Leipzig (Goldhahn et al., 2012). WORTSCHATZ[1] provides randomly crawled, incoherent sentences from various domains (news, web, or `Wikipedia`) in many languages (among many more: German, English, French, Arabic and Russian) indexed by year, and sub-categorized by country (for example, in the case of German, Germany, Austria, or Switzerland). This subdivision allows to select corpora, which are talking about the same, without corresponding to each other.

[1] https://wortschatz.uni-leipzig.de/de/download [Accessed: 6.8.2020]

For this thesis, one million[2] English and German sentences from news of the year 2015 are used as monolingual corpora. The English corpus consists of 180,492 and the German one 378,973 unique words. The huge gap can be explained by the rich German morphology with regard to conjugation and declination (cf. *einen*, *einem*, *einer*, *eine*, *eines* versus *a*) Based upon these, the 500, 1000, and 2000 most common words of each language are extracted to serve as monolingual lexica. To keep the procedure simple, a word is defined as a string of characters between two whitespaces. Words that do not contain characters from the ASCII-Set plus the German-typical *ä*, *ö*, *ü* and *ß* are removed. This does also include digits; despite having the same meaning in English and German, numbers are not part of the lexica, as the small amount of context words (at most 2000) might lead to dissimilar contextual (mis)representations. Terms, which contain punctuation symbols, are split into the parts between. Capitalization is ignored, as well as single-character words, which are grouped as one word. The latter step is done, because it prevents single-letter abbreviations, whose meanings are often ambiguous, from being overrepresented in the vocabulary, while acknowledging their widespread use in the corpora[3]. Both English and German lexica can be found in Appendix A.

There are two reasons for running experiments with inducted dictionaries of increasing size: First, with a growing number of possible words in the context, the distributional representation of each term is expected to improve, and so the translation quality. Second, the increase in the number of states in the FSA is thought to slow down with more words being accepted. Two underlying assumptions justify this claim: On the one hand, the most-common words ought to be easily distinguishable from each other, by sharing a high edit distance. Therefore, in case of the top 500 words, the number of states is expected to be in a similar vicinity. On the other hand, a growing vocabulary makes repeated use of already existing substrings. Thus, the number of states ought to grow slower between the 1000 and 2000 most common words, compared to the increase of states from the top 500 to top 1000 terms.

In conclusion, the quality of the translations is thought to increase with a growing vocabulary, while the growth of the number of states in the FSA constructed on the vocabulary is expected to slowly decrease.

5.1.2 Model Parameters and Experiments

Based on the lexica collected in the last section, word vectors are computed, between which the translation matrix is calculated.

[2]Apparently, the English corpus provides only 965,710 and the German one just 962,330 sentences. Although smaller than declared, this is still a sufficiently large number for a thorough analysis; in total, the corpora comprise of 25,721,910 English and 20,613,618 German words.

[3]It is of course possible to implement exceptions, such as for 'I' and 'a' in English, but this would be contrary to the strict unsupervised approach and is therefore omitted.

5.1.2.1 GLOVE Word Vectors

As already explained, GLOVE is the method of choice. GLOVE (as well as other distributional semantic approaches) offers four main parameters: Context window size, weights on co-occurrences based on the distance from the center term, word vector size, and the number of training iterations. In this project, context size (2, 4, 8, whole sentence) and word vector size (5, 10, 15, 20) are accounted for. As the sentences are randomly crawled and thus not coherent, larger window sizes would contribute to meaning. Although the vector sizes seem very small, it needs to be noted that the lexica only contain around 2000 entries, compared to other publications with vocabularies containing hundreds of thousands of terms. Another benefit is a significantly shorter computation time.
Following the inventors of GLOVE, the distance function is set to $f(n) = \frac{1}{n}$, where n is the distance (in words) from the center term, the number of training epochs is set to 50, which they recommend for vector sizes below 300, and the learning rate η is 0.05. The weighting function (3.49) is applied as suggested, with $x_m ax = 100$, and the exponent being $\frac{3}{4}$.

5.1.2.2 TRANSRANK

Before examining the parameters and experiments in the translation process, the procedure that calculates the similarity between the word vectors is briefly revisited. First, the similarity function $sim(\cdot,\cdot)$

$$sim(x,y) = \frac{1}{1+\mathtt{c}\,\|x-y\|_2} = \frac{1}{1+\mathtt{c}\sqrt{(x-y)^2}} \tag{5.1}$$

receives two matrix entries as arguments, plus a fixed constant c.
Next, the equations for computing the similarity between the matrices $\mathbf{M}$ and $\mathbf{M}'$, which contain the word vectors as rows, are:

$$s_w(v_i,v'_j) = \frac{1-\mathtt{d}}{|V||V'|} + \mathtt{d} \sum_{e_k \in V_e} \sum_{e'_l \in V'_e} \frac{sim(\mathbf{M}[i][k],\mathbf{M'}[j][l])}{\sum_{v_{i'} \in V_w} \sum_{v'_{j'} \in V'_w} sim(\mathbf{M}[i'][k],\mathbf{M'}[j'][l])} s_e(v_k,v'_l) \tag{5.2}$$

$$s_e(e_i,e'_j) = \frac{1-\mathtt{d}}{|V||V'|} + \mathtt{d} \sum_{v_k \in V_w} \sum_{v'_l \in V'_w} \frac{sim(\mathbf{M}[k][i],\mathbf{M'}[l][j])}{\sum_{e_{i'} \in V_e} \sum_{e'_{j'} \in V'_e} sim(\mathbf{M}[k][i'],\mathbf{M'}[l][j'])} s_w(v_k,v'_l) \tag{5.3}$$

These two update rules are applied, until the entries converge.
Taken all this together, three parameters, other than the input matrices $\mathbf{M}$ and $\mathbf{M}'$, can be controlled: Similarity constant c, PAGERANK constant d, and the abort criterion, which discontinues the entries' update.
Throughout the experiments, c is neglected and set to one. Due to the normalization terms

$$\sum_{v_{i'} \in V_1} \sum_{v'_{j'} \in V'_1} sim(\mathbf{M}[i'][k],\mathbf{M'}[j'][l]) \tag{5.4}$$

and

$$\sum_{e_{i'} \in V_2} \sum_{e'_{j'} \in V'_1} sim(\mathbf{M}[k][i'], \mathbf{M'}[l][j']), \tag{5.5}$$

respectively, the similarity between two edge weights is equalized by the sum of similarities between all combinations of outgoing weights. Any change of `c` has to be regarded with respect to this sum, and therefore, `c` is expected to have only minor effects on the overall measurement.

For the damping factor `d`, a value of 0.8 is used, as suggested by Jeh and Widom (2002) and Brin and Page (1998).

As abort criterion, the smallest (at least, in PYTHON3.6) machine-representable number `eps` is chosen, which satisfies[4]

$$1.0 + \texttt{eps} \neq 1.0 \tag{5.6}$$

and is equal to $2.220446049250313 \cdot e^{-16}$. Only, if *all* differences between an updated and its previous entry in the translation matrices are smaller than `eps`, the procedure comes to a halt. This should enforce the program to be ultimately sure in its chosen translation similarities.

To keep the translation process simple, only matrices with the same word vector dimensions are translated into each other. However, this decision is rather driven by computational capacity than theoretical considerations. As the findings of Mikolov et al. (2013) show, word vectors in different languages do not need to correspond in their dimensions to result in an optimal translation. Also, FSA-vectors are only translated into other FSA-vectors; word vectors are translated alike.

In summary, for the 2015 German and English `Wortschatz` news corpora, the proposed unsupervised dictionary induction is evaluated on $(2 \cdot 4 \cdot 4 \cdot 4) = 128$ translation matrices: Two monolingual lexica modes (finite state or conventional monolingual dictionary), in three different sizes (500, 1000, 1500, 2000), based on four different context windows (2, 4, 8, whole sentence[5]), with four different word vector sizes (5, 10, 15, 20).

5.1.3 Evaluation Procedure

The small size of the monolingual lexica unfortunately prohibits any evaluation based on pre-existing gold standard test sets, because the actual semantically closest term or translation might not occur in the monolingual lexica or bilingual dictionary. For instance, the English word *take* could translate to *(mit)bringen*, *brauchen*, *dauern*, *führen*, *(auf)heben*, *(ab, auf, über)nehmen*, *Aufnahme*, *Mitnahme*, *Übernahme* and *tragen*,

[4] https://docs.scipy.org/doc/numpy/reference/generated/numpy.finfo.html [Accessed: 6.8.2020]

[5] By sentence, formally a context size of 100 is meant.

depending on the context. Additional to the infinitive, also the first and second person singular and plural, as well as third person plural form need to be taken into account. In this concrete example, however, only *nehmen, bringen bringt, führen, führt, brauchen, tragen, aufnehmen, übernehmen, Übernahme, dauern* are part of the German lexicon. In the case of the English *national*, the closest German translation would be *international*, as its German equivalent, *national*, does not occur among the most frequent 2,000 words. Handcrafting gold standard similar terms and translations for a total of 4,000 words is, however, also not feasible, as it would exceed the scope of this thesis. Thus, the evaluation is done on a carefully chosen subset of manageable size. Evaluating only a fraction of the data appears at first sight to be too limited, but if chosen reasonable, it can provide a good overview about the framework's performance.
The choice of the words is based on the considerations listed here:

Lexical Units
Only lexical words (which bear, in contrast to functional terms, an actual meaning) are admitted to the data set. Given the size of the data set, words are coarsely categorized into nouns, proper nouns, verbs, adjectives, and adverbs.

Singular & Plural
TRANSRANK's ability to distinguish between singular and plural forms in monolingual lexica and bilingual translation is essential for later MT applications.

Conjugation
In order to evaluate whether different verb forms are differentiated, present, past, and past participle, as well as conjugation forms need to be included.

Declination
Especially in German, nouns and adjectives are inflected depending on their case. Therefore, different case forms occurring in the German lexicon should be integrated to prove that such grammatical information is incorporated into the word vectors.

Grammatical Gender
Following this rationale, grammatical gender is also tested.

Morphological Derivation
Another important point is morphological derivation. The question is whether TRANSRANK is able to grasp derived meanings, for instance recognizes the agent of an verb, or the adjectival description of a noun.

Semantics
Included for formal reasons, the question to be answered is, how much information beyond the grammatical aspects are incorporated. Test cases include two basic semantic relations, antonymy and entailment.

OOV-Terms Last but not least, the performance on out-of-vocabulary words is taken into account. How accurate is the translation process, if the counterpart is not present in the target language?

In order to test these aspects properly, irregular forms are included; doing so ensures that the actual *semantic* information, for instance plural form, is captured, rather than the mere presence of certain morphemes, such as ′-s′. For brevity, the selection of words for the test sets are listed in Appendix A (English) and Appendix B (German). While the English test set comprises of 47 lemmata with 113 word forms, the German set contains the same number of lemmata, with 116 word forms. Due to the almost absence of case markers in English, only singular and plural forms are included into declination. The words *haven, hasn, doesn, don, didn* are the negated forms of *have, has, does, do,* and *did*, because the tokenization procedure cuts *′t* apart. Tables 5.1 and 5.2 list the test questions for English and German.
Since the similarity of two words is at least to some extend based on subjective perception, and rather a gradual than binary feature, the test cases are constructed to be as unique as possible, in the form of w_1 *to* w_2 *is like* w_3 *to* $w_?$. Plain similarity results are calculated only for the three OOV terms, where the judgement is left to the reader.

Test Case	Frequency Rank		
	≤500	≤1000	≤2000
Gender			man:woman::boy:girl men:women::boys:girls
Declination	day:days::year:years man:men::woman:women country:countries::year:years		player:players::driver:drivers woman:women::girl:girls man:men::boy:boys
Derivation	America:American::Europe:European	play:player::drive:driver America:American::Russia:Russian	play:players::drive:drivers move:movement::think:thought move:movement::help:helping increase:increasingly::report:reportedly move:movement::feel:feeling develop:development::move:movement move:movement::think:thought
Conjugation	take:taking::go:going take:took::go:went take:took::have:had take:taken::have:had go:going::have:having have:has::do:does have:having::do:doing do:doing::make:making do:did::make:made do:done::make:made	take:takes::go:goes show:showed::play:played show:shows::go:goes	go:gone::take:taken show:showing::do:doing have:haven::do:don have:hasn::do:doesn play:plays::do:does
Comparison Distinction		high:higher::low:lower good:better::high:higher good:best::large:largest	bad:worst::high:highest good:well::large:largely better:best::higher:highest larger:largest::higher:highest bad:worse::good:better worse:worst::better:best high:highly::large:largely
Antonymy Entailment		high:low::large:small good:bad::high:low	America:Washington::Russia:Moscow higher:lower::larger:smaller England:London::France:Paris America:Washington::England:London boy:man::girl:woman men:boy::women:girls

TABLE 5.1: List of English Questions for Evaluation

Test Case	Frequency Rank		
	≤500	≤1000	≤2000
Gender			Mann:Frau::Junge:Mädchen Jungen:Mädchen::Männer:Frauen
Declination	Tag:Tage::Jahr:Jahre Tag:Tagen::Jahr:Jahren	Länder:Ländern::Tage:Tagen Länder:Ländern::Jahre:Jahren Mann:Männer::Junge:Jungen Mann:Männer::Frau:Frauen	Smartphone:Smartphones::Tag:Tage Smartphone:Smartphones::Jahr:Jahre Tag:Tages::Jahr:Jahres Europa:Europas::Russland:Russlands Frau:Frauen::Mädchen:Mädchen
Derivation		spielen:Spieler::fahren:Fahrer Russland:russische::Europa:europäische Russland:russischen::Europa:europäischen	helfen:Hilfe::denken:Gedanken öffnen:offen::erhalten:erhältlich bewegen:Bewegung::fühlen:Gefühl bewegen:Bewegung::helfen:Hilfe fühlen:Gefühl::denken:Gedanken
Conjugation	gehen:geht::haben:hat gehen:ging::haben:hatte zeigen:zeigt::machen:macht	nehmen:nimmt::spielen:spielt nehmen:genommen::zeigen:gezeigt gehen:gehe::haben:habe tun:tat::machen:machte	nehmen:nahm::spielen:spielte gehen:gegangen::spielen:gespielt ging:gingen::hatte:hatten hatte:gehabt::tat:getan gehen:gingen::machen:machten gehen:gehe::machen:mache tun:tut::haben:hat
Comparison Declination		groß:große::gut:gute große:kleine::großen:kleinen große:großen::größte:größten gute:beste::große:größte guten:besten::großen:größten	groß:großes::gut:gutes kleinen:kleiner::großen:großer gut:guter::groß:großer besser:bessere::größer:größere
Antonymy Entailment		gut:schlecht::große:kleine gut:schlecht::großen:kleinen	Russland:Moskau::USA:Washington USA:Washington::England:London England:London::Frankreich:Paris Mann:Junge::Frau:Mädchen Jungen:Männer::Mädchen:Frauen gut:schlecht::großer:kleiner

TABLE 5.2: List of German Questions for Evaluation

As done by Mikolov et al. (2013b), the calculation is carried out by

$$\mathbf{w}_? = \mathbf{w}_2 - \mathbf{w}_1 + \mathbf{w}_3 \tag{5.7}$$

with the actual word w_4 being the closest result to vector $w_?$:

$$w_4 = \arg\max_{w_?} \; cos(w_2 - w_1 + w_3, w_?). \tag{5.8}$$

The test set for the evaluation of the translations is compiled by the lemmata the German and English test set list have in common. Since the quality of the translations is yet unknown, in this first evaluation, all forms of a lemma are considered being valuable. It could be that the highest ranked translation for *tun* (*(to) do*) is *didn*, which would be accepted as correct. For the moment, the translation quality must therefore be viewed as an upper bound for the approach.

There are several options for evaluation metrics. Classic translation quality measures, such as BLEU (Papineni et al., 2002) or ROUGE (see Lin and Och (2004) and Lin (2004)) do not apply, because they are designed to compare substrings, instead of single words. Therefore, a straight-forward precision/ recall statistic, as employed by WORD2VEC, GLOVE and FASTTEXT, would be fully sufficient: Precision p denotes the number of correctly identified closest analogous words or translations by a system, divided by all *detected* analogies or translations, respectively; recall r is the number of all correctly identified closest analogous terms, or translations, in relation to all *possible* analogies or translations in the lexica. $F1$ combines both measures in one:

$$F1 = 2 \cdot \frac{p \cdot r}{p + r} \tag{5.9}$$

A tempting alternative is the *relative rank*, which is used by Dorow et al. (2009), as it reflects uncertainty about the quality. Simply providing the precision and recall scores for the closest results might not give a good overview of the performance, if the second or third best vector is the desired outcome. The relative rank bypasses this disadvantage:

$$r(w) = \frac{\text{rank}(w)}{\mid \text{Vocabulary} \mid} \tag{5.10}$$

where

$$0 \leq \text{rank}(w) < \mid \text{Vocabulary} \mid \tag{5.11}$$

is the rank of a desired outcome w; zero denotes the top, $\mid$ Vocabulary $\mid -1$ the bottom rank. This yields for $r(w)$:

$$0 \leq r(w) \leq \frac{\mid \text{Vocabulary} \mid -1}{\mid \text{Vocabulary} \mid} \tag{5.12}$$

Doing so accounts for the whole range of outcomes and allows a better comparability of the effects of context size, state or word embedding, and embedding dimensions on vocabulary size. This is why, it is also used to evaluate this project. For a better visibility in graphs, the relative rank is subtracted from one, such that an outcome of one represents the best possible result.
Another rank-based measurement that used by related work (Rothe and Schütze, 2014), is the *mean reciprocal rank* (hence, MRR) (Voorhees and Tice, 1999):

$$\text{MRR} = \frac{1}{n} \sum_{i=0}^{n} \frac{1}{rank(w_i)} \tag{5.13}$$

MRR denotes the sum of inversed ranks of the correct answers. Doing so yields a "a more fine-grained measure" than precision among the top k answers (Laws et al., 2010).

The evaluation procedure is, just as the experimental setup, twofold: First, the quality of the monolingual word vectors is tested, and second, basing on those, the translation matrices are evaluated.

5.2 Results and Comparison to Related Work

This section presents the results for proposed framework. First, the outcomes of the GLOVE vectors are shown, then the ones of TRANSRANK.

5.2.1 Word Vectors

The first hypothesis to be tested is the manageable number of states compared to the number of words encoded by them. As stated in Section 3.3, not all but only a subset of states needs to be considered. Plots in Figure 5.1 contrast the number of words, states, and *relevant* states for English (left) and German (right):

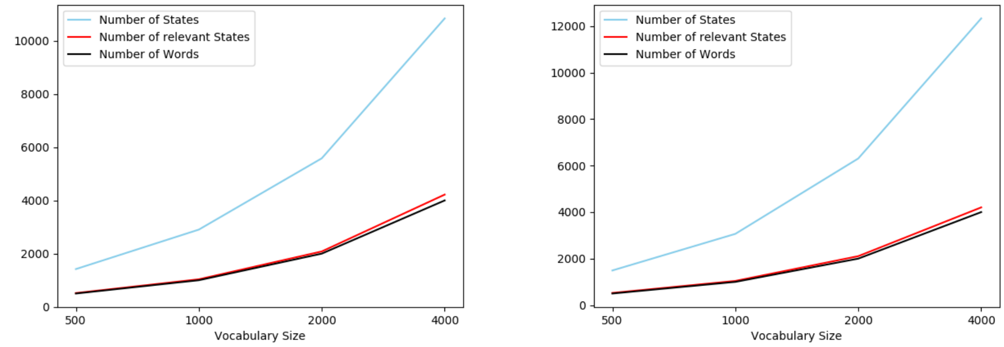


FIGURE 5.1: Development of the Number of States and Words for English (left) and German (right)

The vocabularies consisting of the most common 4,000 words is not included in other tests; it is merely compiled for this chart to emphasize that with every duplication of the vocabulary size, the number of relevant states stays roughly in the same magnitude as the number of words. In case of the actual number of states, the German FSA contains more states, probably because German words tend to be longer than English ones. Regarding the *relative* growth, the next plot depicts the first derivative of above numbers. Doing so can help to identify trends for larger vocabularies:

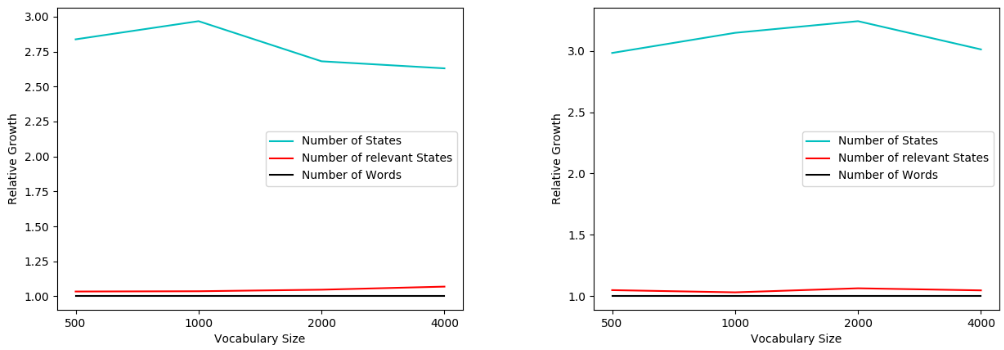


FIGURE 5.2: Relative Development of the Number of States and Words for English (left) and German (right)

In case of the number of relevant states, there is no consistent relative behavior for English and German. While increasing for the former, it decreases for the latter during the last redoubling from 2000 to 4000 words. Interestingly, the development of the total number of states declines rapidly both for German and English. This means, lesser states are added, but more are re-combined by newly introduced transitions, leading to more *decisive*, i.e. relevant states.

Before diving into a detailed analysis of all parameters, the first overview on the top five similar words for the six OOV terms showcase the drawbacks of the overall approach:

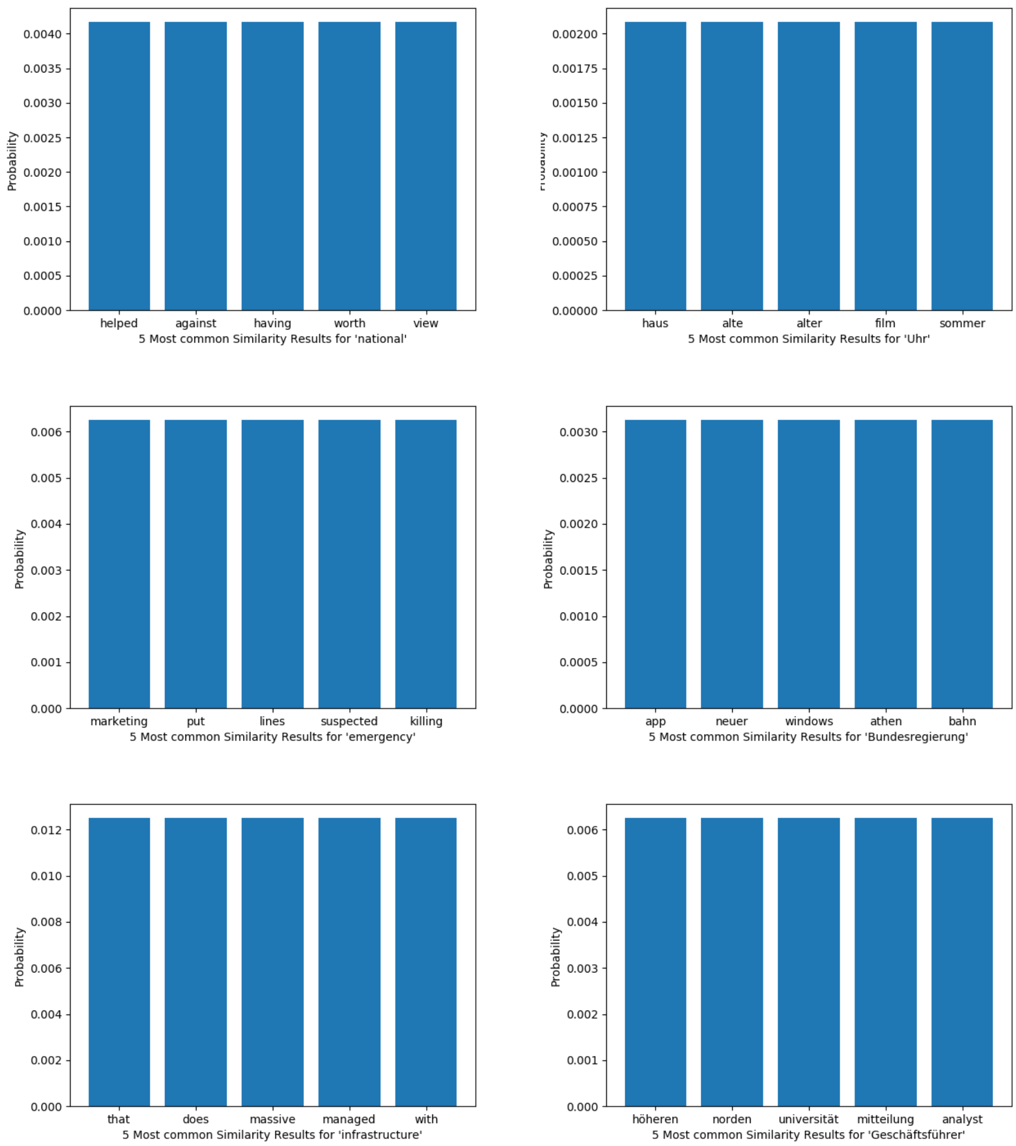


FIGURE 5.3: Most common top 5 similar Results for OOV Terms in English (left) and German (right)

Figure 5.3 shows the most common outcomes among the top five similar words for the OOV terms (ignoring state-/ word embedding, vector dimensions, context and vocabulary size) together with their probability. The English translations for the German words are: *Uhr - clock*, *haus - house*, *alte - old*, *alter - old/ age*, *film - movie*, *sommer - summer*, *Bundesregierung - federal government*, *app - app*, *neuer - new(er)*, *windows - windows*, *athen - athens*, *bahn - train/ lane*, *Geschäftsführer - general manager*, *höheren*

- *higher*, *universität - university*, *mitteilung - message*, and *analyst - analyst*. Beyond maybe metaphorical similarity, the results share no common semantic ground. Also, their probabilities are equal, meaning that vectors of the most common results cannot be further distinguished. Next, the probability distribution over the first top ten most probable answers to the analogy questions shed light on a problem present throughout the evaluation:

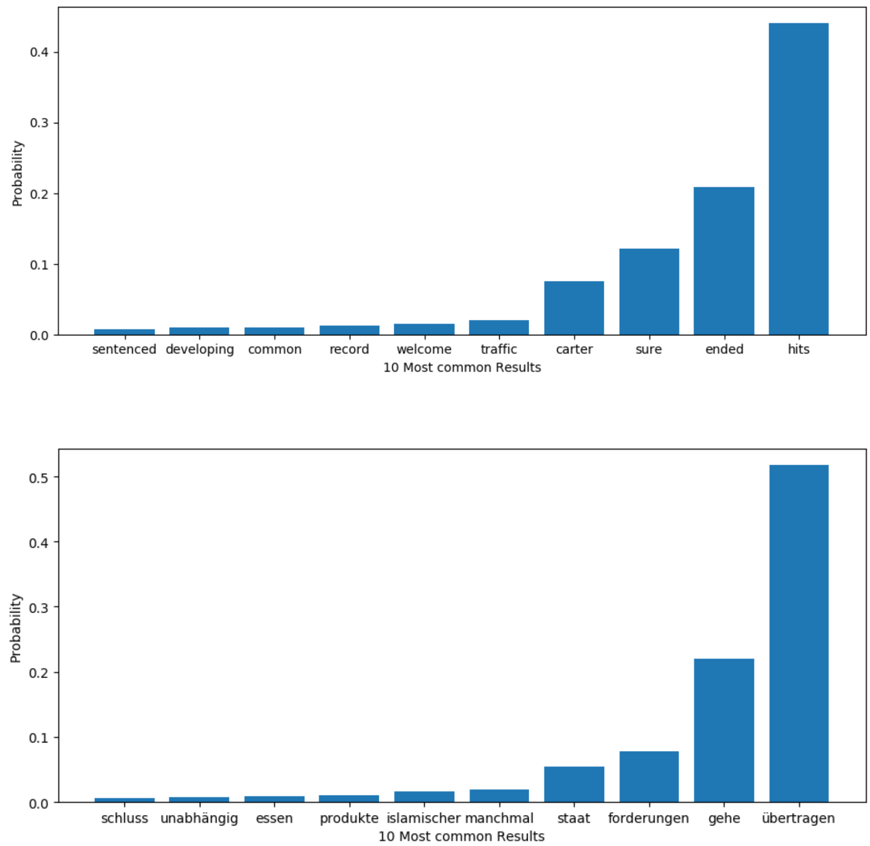


FIGURE 5.4: Top 10 English (above) and German (right) Answers

The English translations are *schluss - end*, *unabhängig - independent*, *essen - (to) eat/food*, *produkte - products*, *islamischer - islamic*, *manchmal - sometimes*, *staat - state*, *forderungen - claims*, *gehe - (I) go*, *übertragen - (to) convey*. Here, the actual meaning is negligible; it is important to note that in more than 40% (50%, respectively) of all questions, the same answer is given. The following graphs break the answers down to the different vocabulary sizes:

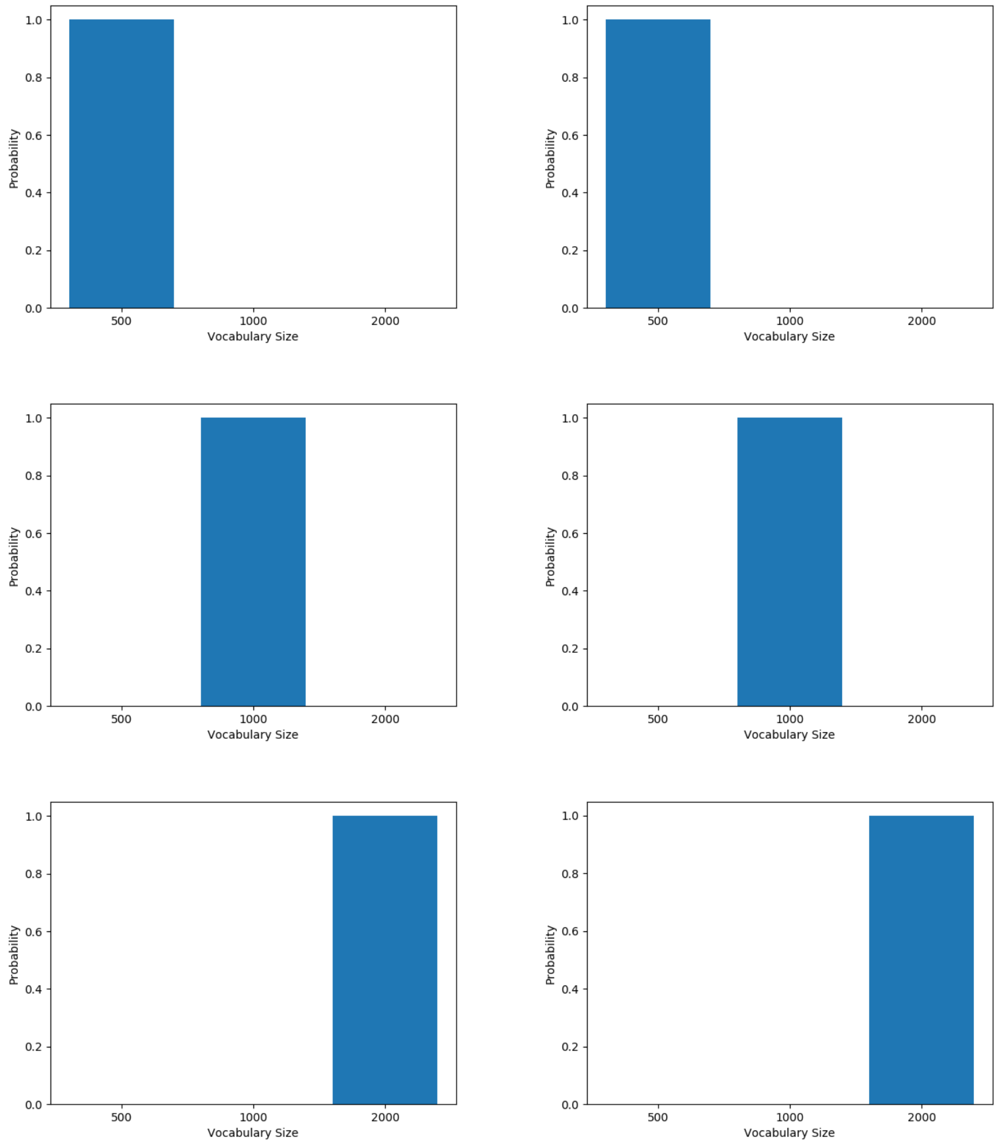


FIGURE 5.5: Distribution of Results per Vocabulary Level for English (left) and German (right) questions. From top to bottom: Most common 500, 1000 and 2000 words.

It can be seen that the answers for questions of a certain vocabulary size are (almost) exclusively from that very vocabulary. Furthermore, *hits*/*übertragen*, *ended*/*gehe* and *sure*/*staat*, have frequency ranks *500*, *1000* and *2000*, meaning, those words are the least frequent words of the top 500/1000/2000 vocabulary.
Until now, the results are jointly analyzed across word and state embeddings, context and embedding sizes. The upcoming graphs investigate the influence of those parameters. For a concise visual presentation, box-plots from `Pythons` MATPLOTLIB[6] are used. Box-plots characterize a distribution of ordered data points by five essential values: The median, the upper and lower quartile, and two whiskers, which are

[6] https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.boxplot.html [Accessed: 6.8.2020]

by default defined by the closest data points to the interquartile range times 1.5 plus/ minus the upper and lower quartile. Data points outside these margins are considered as outliers. The median shows which value is larger than 50% of the data, and the size of the interquartile range, indicated by a box, illustrates how skewed the distribution is towards the upper or lower end of the scale. Multiplying this range by 1.5 is a well established convention, which dates back to Tukey (1977).

There also exist other techniques for data exploration, which measure interactions between several variables, for example regression models. However, as the first results turn out to be less than ideal, box plots capture and picture the coarse effects of the various parameters well, while for starters skipping too fine-grained details, which might only be a matter of coincidence in the small data set. A deeper analysis of how parameters are interconnected is suggested for future work.

Figure 5.6 shows the relative ranks for word embeddings for the three vocabulary sizes (top row 500, middle row 1000, and bottom row 2000 words) for English (left) and German (right-hand side plots), given the four different context sizes:

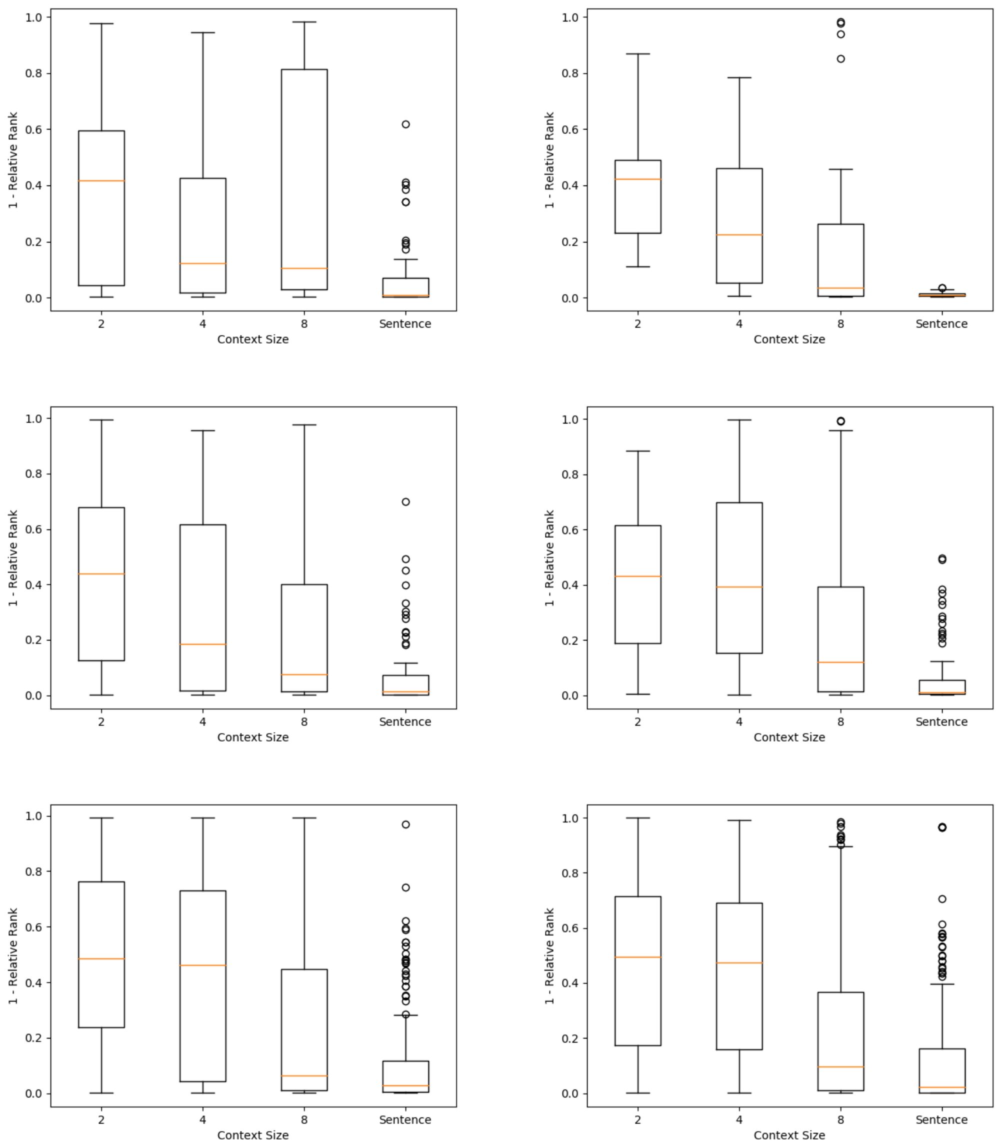


FIGURE 5.6: Influence of Context on English (left) and German (right) Word Embedding Responses to Analogy Questions. From top to bottom: Most common 500, 1000, 2000 Words.

The next chart shows the same subject matter for state embeddings:

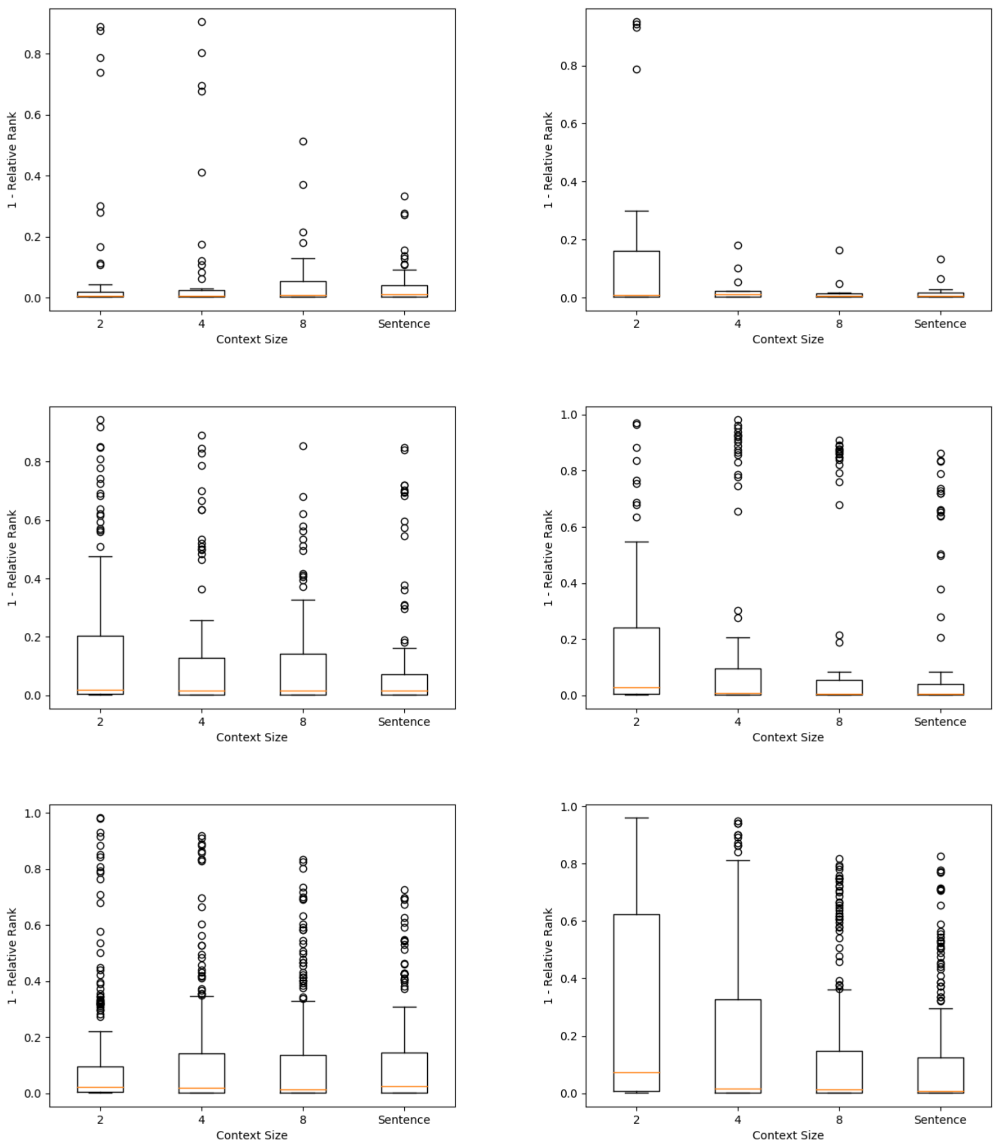


FIGURE 5.7: Influence of Context on English (left) and German (right) State Embedding Responses to Analogy Questions. From top to bottom: Most common 500, 1000, 2000 Words.

The first thing to be noticed are the low medians; in the best cases, 50% of the correct answers are ranked in the lower half of *all* possible outcomes. Less formally, half of the vocabulary is tried before giving the right answer. For both languages, a small window of two yields the best results. With a growing vocabulary, the median relative rank with a context of four words is on par with a window of two, however, its second lower quartile (25% to 50% of the ascending ordered data) has a wider range of values. State embeddings rate significantly worse than word embeddings. For German, the FSA-approach works better, which is indicated by higher medians and less outliers. This is probably due to the richer German morphology. In all setups, a sentence-wide context window seems not favorable.
Another parameter is the embedding dimension. Four dimensions are tested, 5, 10,

15, and 20. As with the last plots, the right-hand side graphs display the results for English, and the left-hand side ones for German. The vocabulary size is again increasing from top (500 words) to bottom (2000 words). First, the results for word embeddings:

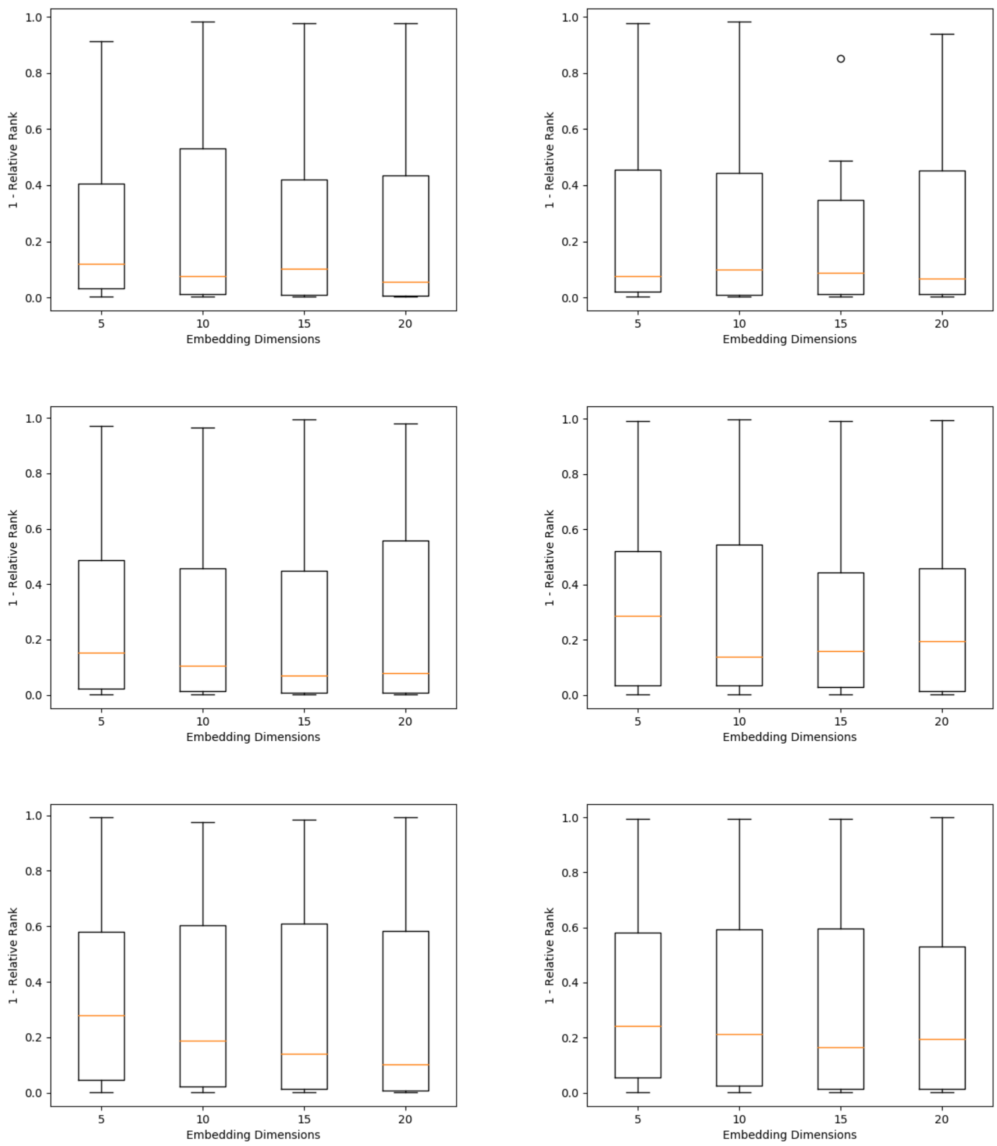


FIGURE 5.8: Influence of Vector Dimensions on English (left) and German (right) Word Embedding Responses to Analogy Questions. From top to bottom: Most common 500, 1000, 2000 Words.

The relative ranks for state embeddings are given in the following plots:

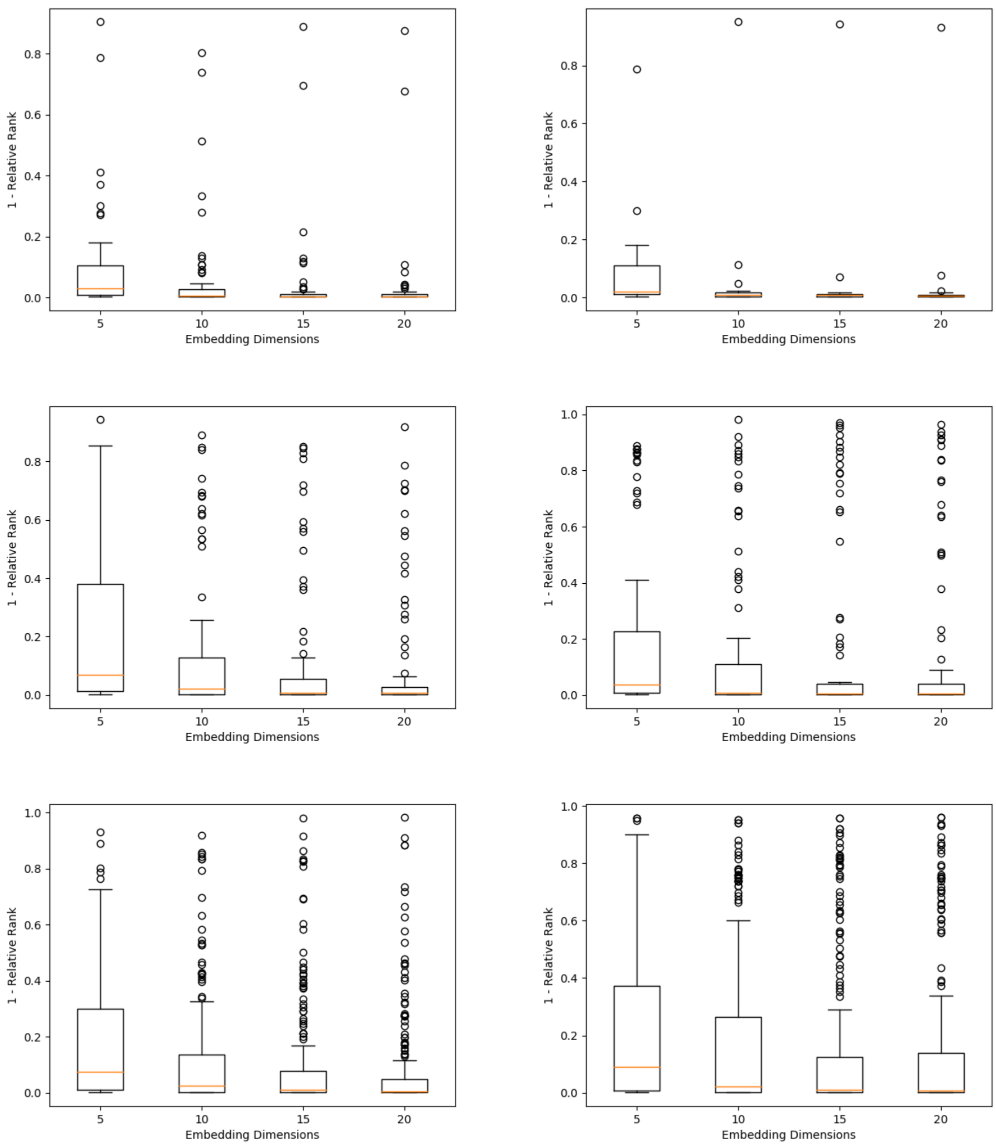


FIGURE 5.9: Influence of Vector Dimensions on English (left) and German (right) State Embedding Responses to Analogy Questions. From top to bottom: Most common 500, 1000, 2000 Words.

Both figures 5.8 and 5.9 reveal a clear tendency towards low embedding dimensions. The next plots investigate differences in the results between the types of questions. Firstly, with word embeddings:

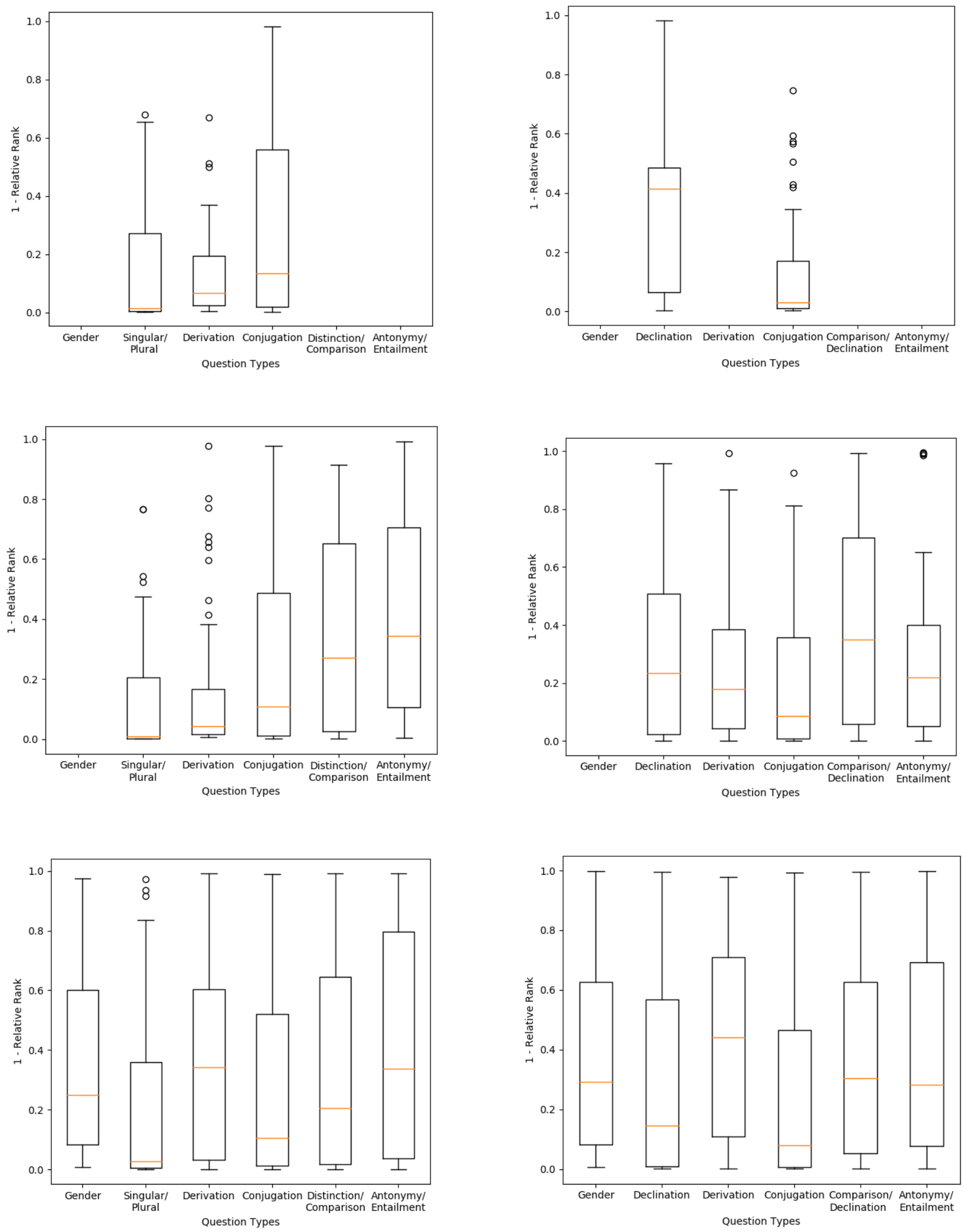


FIGURE 5.10: Comparison of Results for Question Types of English (left) and German (right) Word Embeddings. From top to bottom: Most common 500, 1000, 2000 Words.

Secondly, for state embeddings:

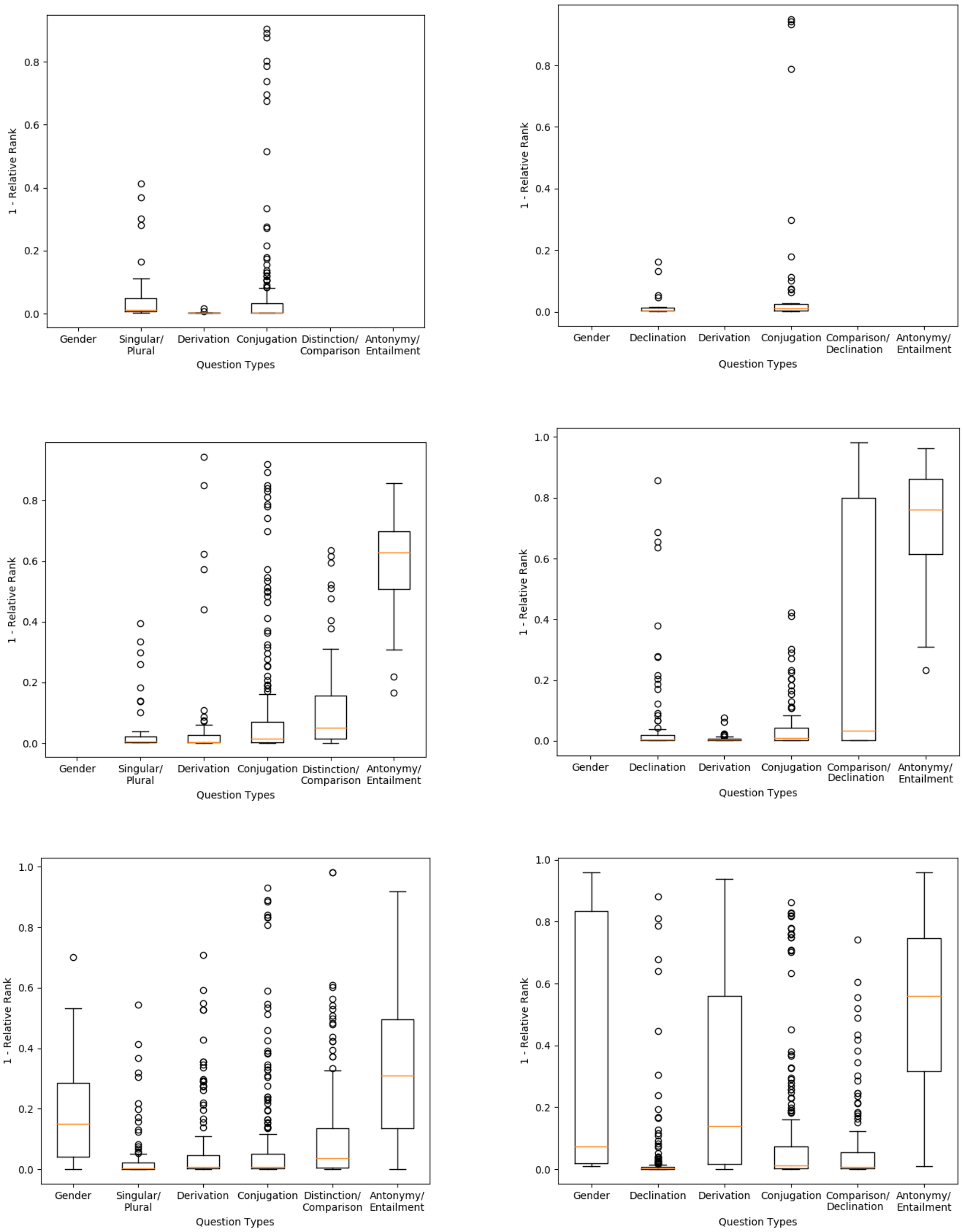


FIGURE 5.11: Comparison of Results for Question Types of English (left) and German (right) State Embeddings. From top to bottom: Most common 500, 1000, 2000 Words.

Lower vocabulary sizes omit some question types, because the terms are not part of the dictionary. Result patterns for English and German are quite comparable. The main difference is the low median for singular/ plural questions in English and the declination questions in German. Highest rated answers are for derivative questions, followed by entailment and adjectival comparison/ distinction questions. State embeddings lead in general to significantly lower ranks, with one notable exception: semantic analogies are answered as good in English and much better in German.

In order to get an idea of the best on-average performance of the vectors, the highest-scoring ⟨context size, embedding mode, embedding size⟩ parameter triplet for each vocabulary size is selected. Table 5.3 shows those settings for English:

Vocabulary Size	Context Size	Embedding Mode	Embedding Size	Average Relative Rank[7]	Average Rank[8]
500	2	Word	10	≈0.51187	245
1000	2	Word	20	≈0.57200	429
2000	2	Word	10	≈0.54102	918

TABLE 5.3: Best on Average Parameter Settings for English Vectors

And similarly, Table 5.4 those for German:

Vocabulary Size	Context Size	Embedding Mode	Embedding Size	Average Relative Rank[9]	Average Rank[10]
500	2	Word	20	≈0.51433	243
1000	2	Word	10	≈0.47561	525
2000	2	Word	10	≈0.55177	897

TABLE 5.4: Best on Average Parameter Settings for German Vectors

Both tables emphasize that word embeddings clearly outperform state embeddings, and small window sizes work best for both languages. However, *outperform* must be understood in relation to the other parameter combinations: Even the highest achievable relative ranks come close to guessing, since the expected rank of a uniform distribution over all relative ranks amounts to 0.5.
Finally, before continuing with the translation results, another hypothesis is tested. Do results for questions from lower vocabularies improve in larger vocabularies, and if so, how much? The upper plots display English, and the lower ones German questions. On the left, only results for questions from the 500 words test set are shown; the plots to the right display results for questions from the top 1000 words. Again, the results for word embeddings are initially presented:

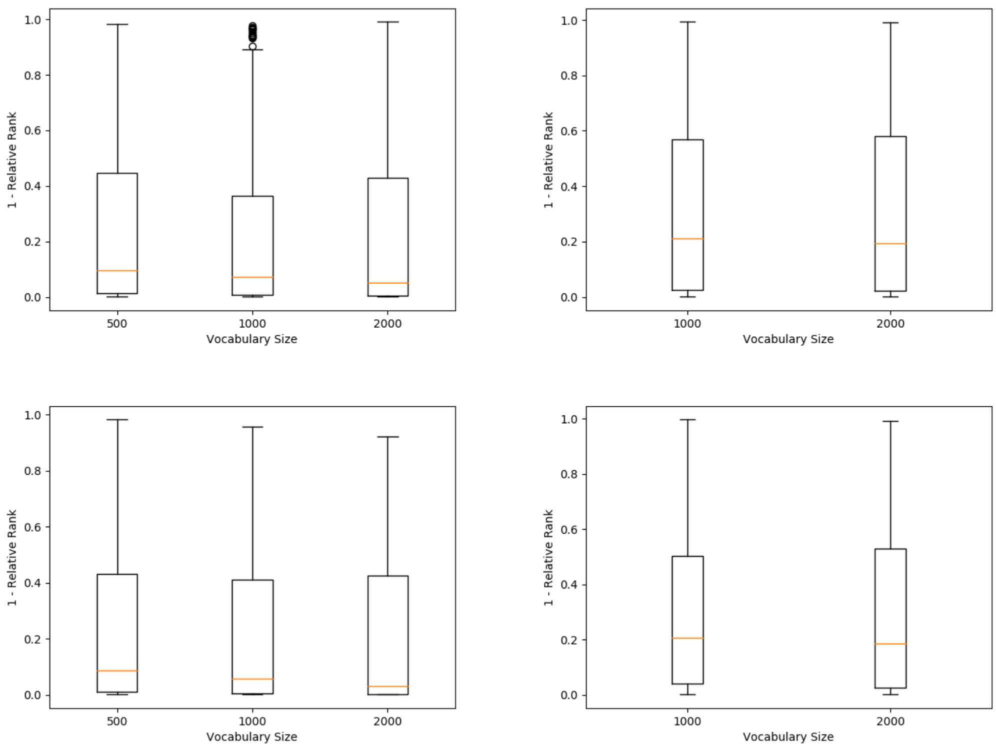


FIGURE 5.12: Differences of Results for Questions from lower Vocabulary Sizes with Word Embeddings. Questions from the top 500 (left) and top 1000 words (right).

Results for state embeddings are as follows:

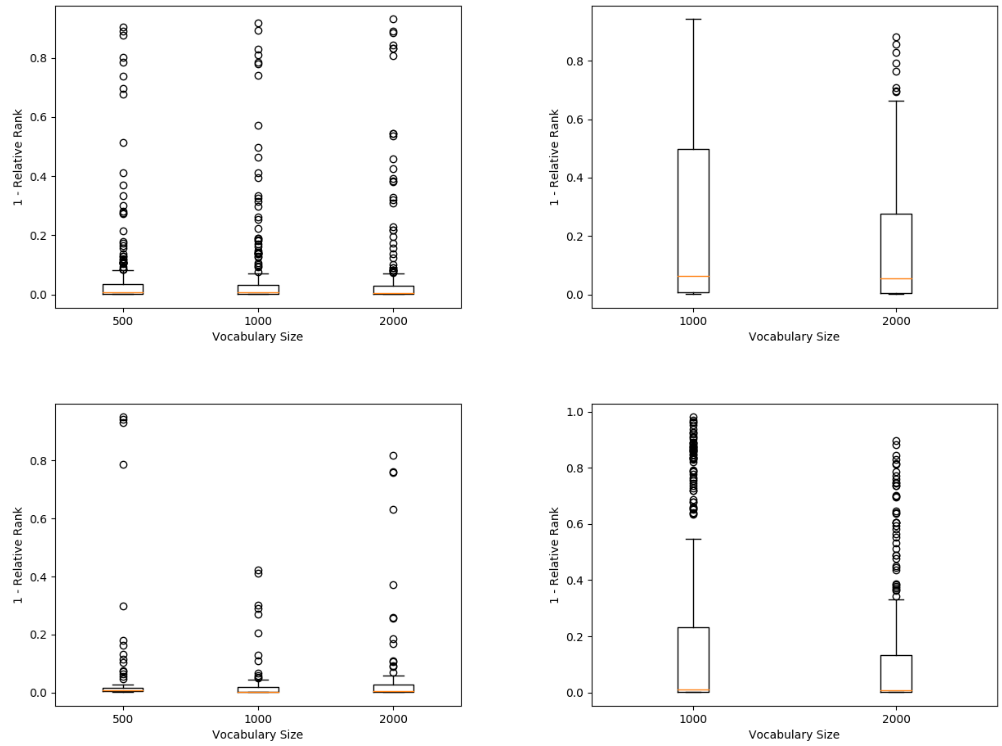


FIGURE 5.13: Differences of Results for English (left) and German (right) Questions from lower vocabulary sizes with State Embeddings. Questions from the top 500 (left) and top 1000 words (right).

Contrary to the expectation, results for questions of lower vocabulary sizes do neither improve for word nor state embeddings, if the number of words increases. A suitable explanation for this phenomenon could not be found.

Drawing comparisons to the word vector approaches presented in Section 3.2.2 is rather complicated. Their accuracy based evaluations are hardly comparable to the results shown in this section, because the desired answer never appeared on first rank. A strict accuracy measure would therefore yield zero percent of correctly given answers. GLOVE reaches an accuracy of 69.3% on semantic and 81.9% on syntactic questions, CBOW and Skip-gram score 68.9%/ 57.3% and 65.1%/ 66.1% on semantic/ syntactic analogies, respectively (cf. Table 3.8/ GLOVE).

The situation for state embeddings is worse. Originally meant to be an alternative to FASTTEXT, state embeddings do not even replicate those results to at least some extent. On German semantic and syntactic questions, FASTTEXT achieves 62.3% and 56.4%, on English, it gained 77.8% and 74.9% (cf. Table 3.10, FastText).

This is not only the case in analogical questions, as demonstrated by the top five most similar words for OOV terms shown in Figure 5.3, and the distribution of the most common outcomes (cf. Figure 5.4). In summary, it seems that embedding vectors computed for this thesis are not able to grasp relevant information on semantic and syntactic level.

5.2.2 TRANSRANK

Provided with the results from the word vectors' evaluation, the actual analysis on the interlingual alignments is presented.
The first part of the evaluation is concerned with the convergence of the translations matrices. The longer it takes until the abort criterion is met, the more indifferentiable the entries of the embedding matrices are, and thus the more difficult the disambiguation of the embeddings is. It is therefore expected that especially small vocabulary sizes, low embedding dimensions and a context too large cause slower convergence. For the same reason, translations between state embeddings are hypothesized to take longer to converge. Subsequently, the plots show the influence of context size on convergence for word (left) and state (right) embeddings and the most common 500 (top), 1000 (middle) and 2000 (bottom) words:

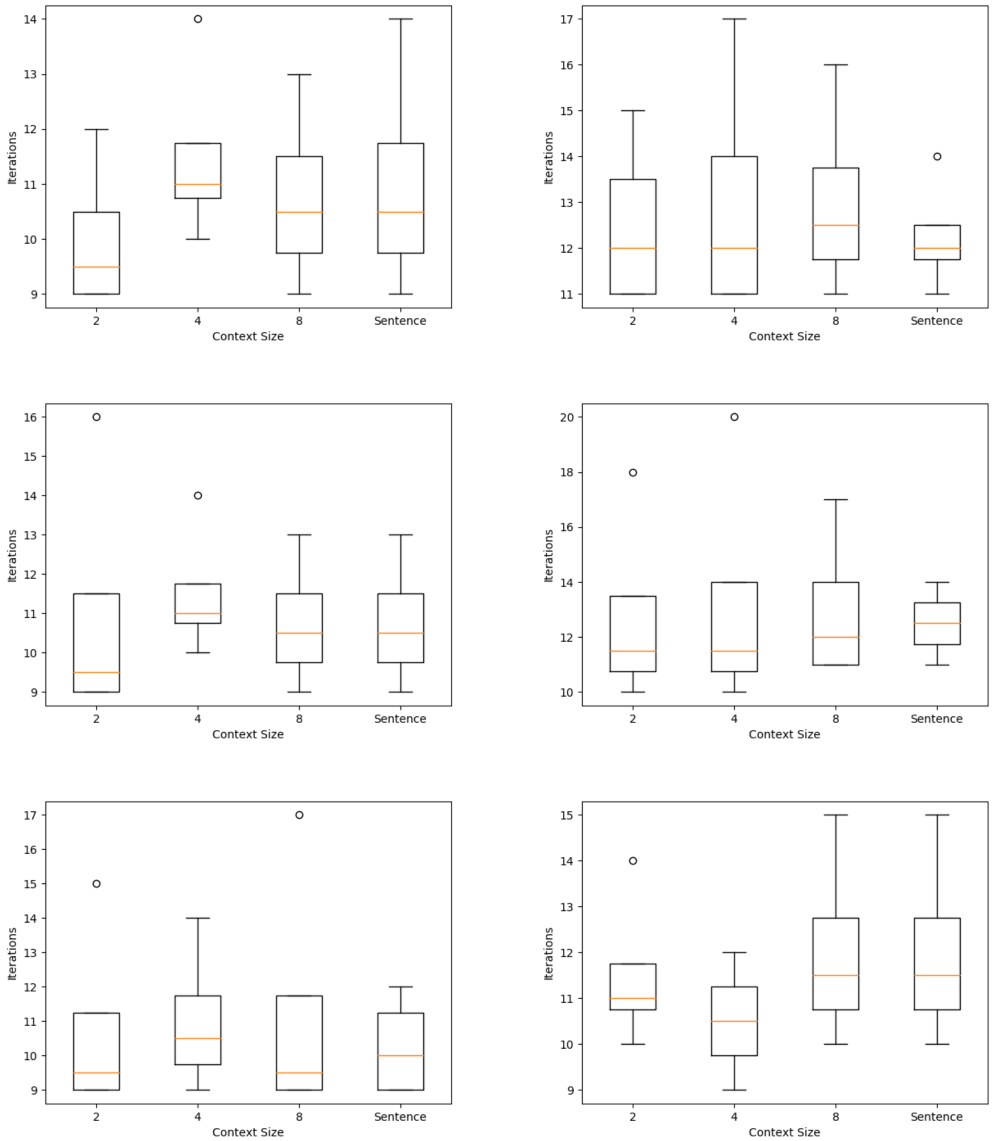


FIGURE 5.14: Convergence Rates for Context Sizes for Word (left) and State (right) Embeddings. From top to bottom: Most common 500, 1000, 2000 Words.

While taking the whole sentence as context into account leads throughout to a high number of iterations, a context of two four results also in a slow convergence. Using only the preceding and subsequent word seems to be the optimum in terms of efficiency.

In the fashion of Figure 5.14, the upcoming chart shows the effect of embedding dimensions:

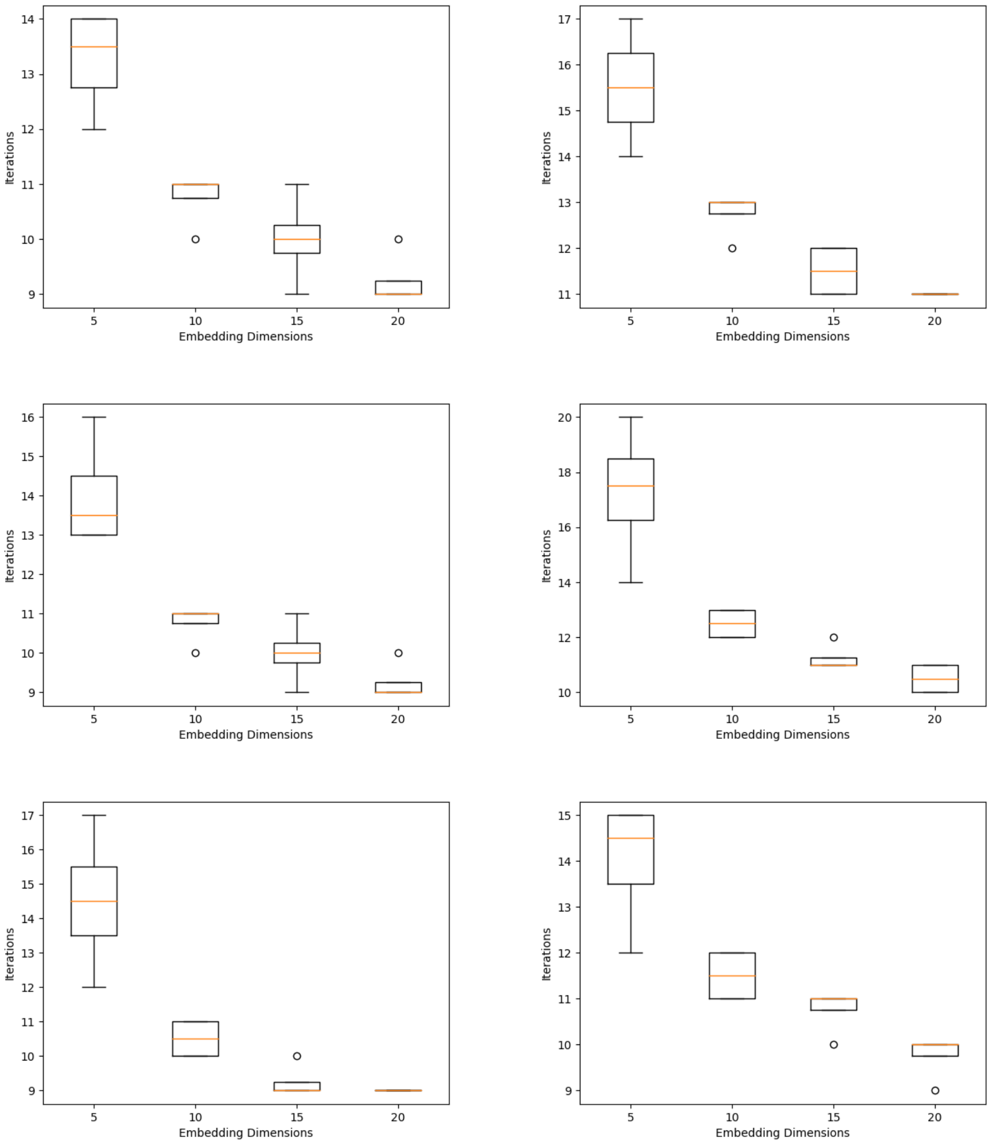


FIGURE 5.15: Convergence Rates for Embedding Dimensions for Word (left) and State (right) Embeddings. From top to bottom: Most common 500, 1000, 2000 Words.

Clearly, convergence takes significantly longer for a small number of embeddings. Overall, it can be stated that with a larger vocabulary, convergence rates decrease, and that state embeddings need usually more iterations than word embeddings.
In order to get a more intuitive overview of the system's abilities, the first plots show the top ten results of German-to-English (left) and English-to-German (right) translations.

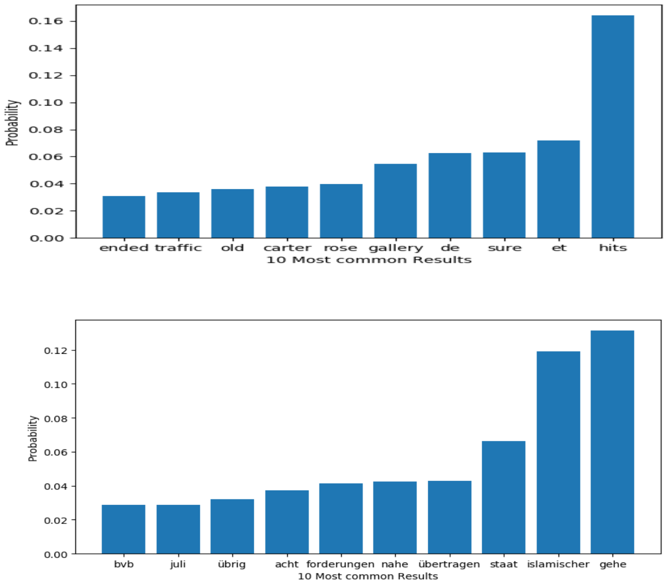


FIGURE 5.16: Top 10 English (left) and German (right) Translations

Translations for the German words are *gehe - (I) go*, *islamischer - islamic*, *staat - state*, *übertragen - (to) transfer*, *nahe - near*, *forderungen - claims*, *acht - eight / attention*, *übrig - left*, *juli - july*, *bvb - bvb (German soccer club)*. However, more important than the actual translations is their probability: Compared to the most common outcomes of the analogy questions (cf. Figure 5.4) the curves are much flatter, meaning, the set of possible outcomes is much more variant than for word vectors. The next figure presents the five most common translations for the OOV terms:

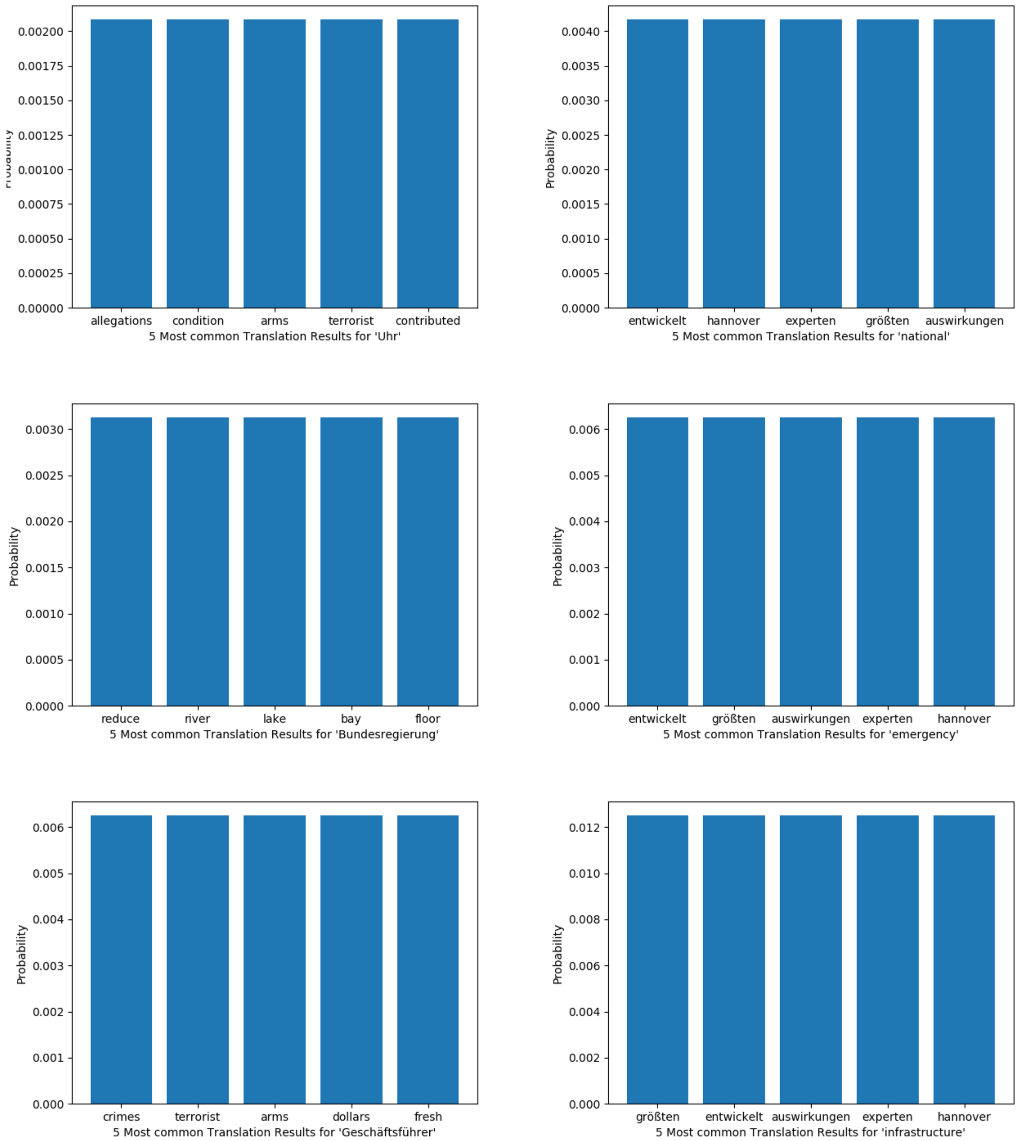


FIGURE 5.17: Most common top 5 similar Results for German (left) and English (right) OOV Terms

English equivalents for the closest German translations are *entwickelt - developed/ develops*, *hannover - Hannover (German City)*, *experten - experts*, *größten - biggest*, *auswirkungen - effects*. None of those resulting top-translations is related to the source words. Even worse, as in the case of the top five most similar words within the respective languages (see Figure 5.3), the translations are all equally likely, meaning there is no distinction among the most common translations, although they are unrelated.
Next, the distribution of the computed goal words over the vocabulary levels is investigated:

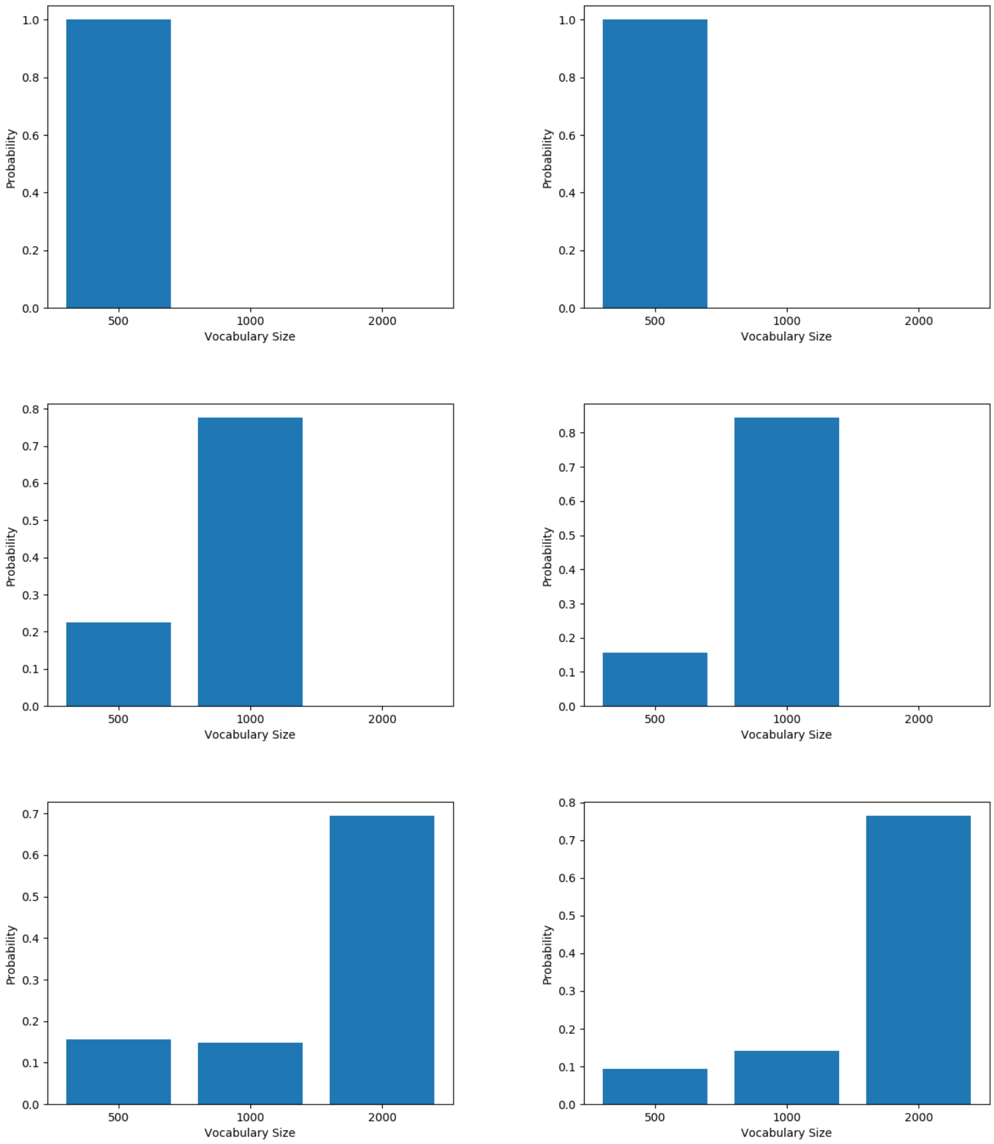


FIGURE 5.18: Distribution of Results per Vocabulary Level for German-to-English (left) and English-to-German (right) Translations. From top to bottom: Most common 500, 1000 and 2000 words.

Other than for monolingual word vectors (see overview in Figure 5.5), also terms from smaller vocabularies are selected, albeit to a lesser extend.

The upcoming graphs present the effects of context size, embedding dimensions, and the PoS-tag of the source word on the relative rank of the 'correct' translation. At this point, the reader is reminded that for simplicity, only monolingual word vectors of the same context and embedding sizes are translated into each other. Also, the relative rank is calculated for the highest-ranked word-form of all existing types of the correct lemma in question.

Figure 5.19 shows the influence of context on the relative rank for word vectors. The outcomes for German-to-English are on the left, those for English-to-German on the right; the vocabulary sizes are ascending from top to bottom.

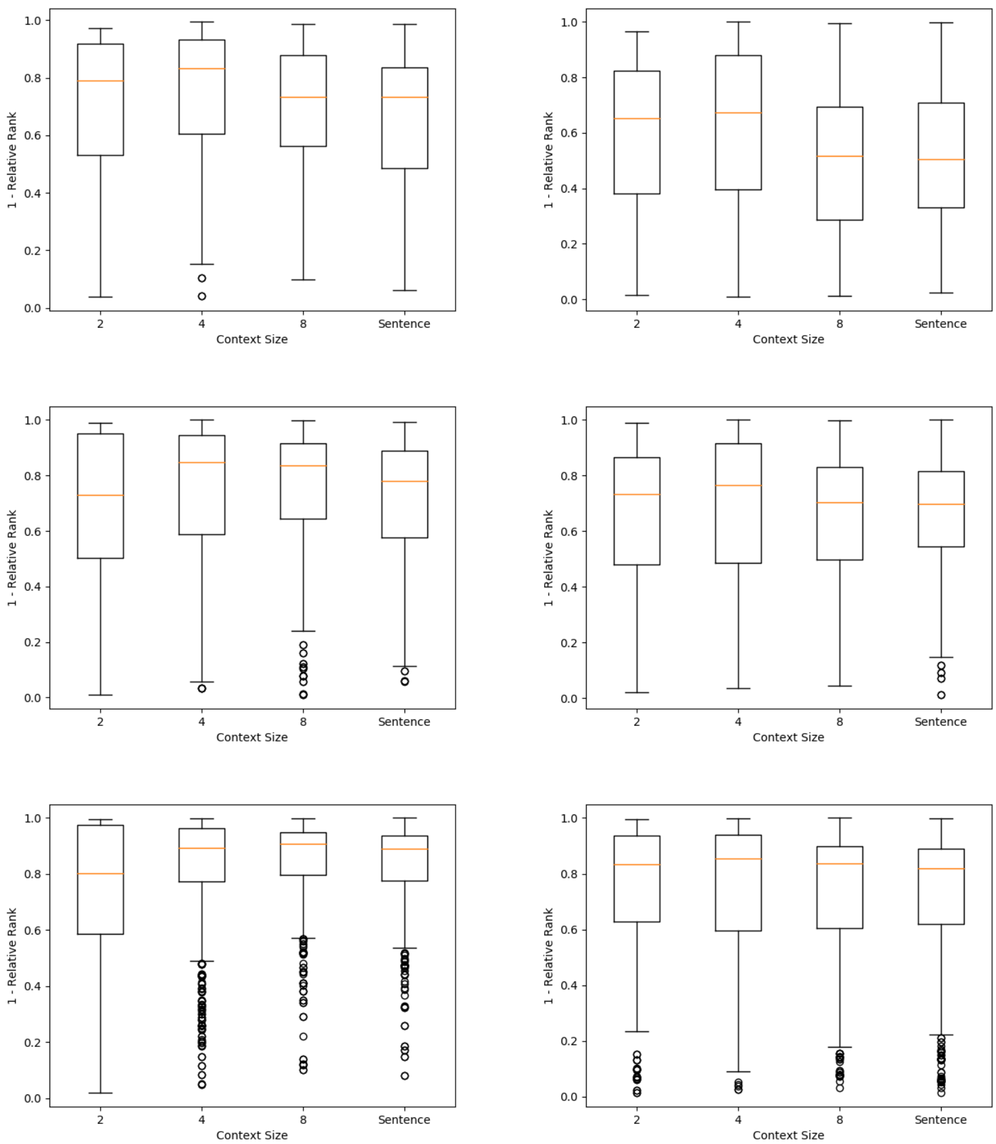


FIGURE 5.19: Influence of Context on German-to-English (left) and English-to-German (right) Word Embedding Translations. From top to bottom: Most common 500, 1000 and 2000 words.

Likewise, the results for state embeddings:

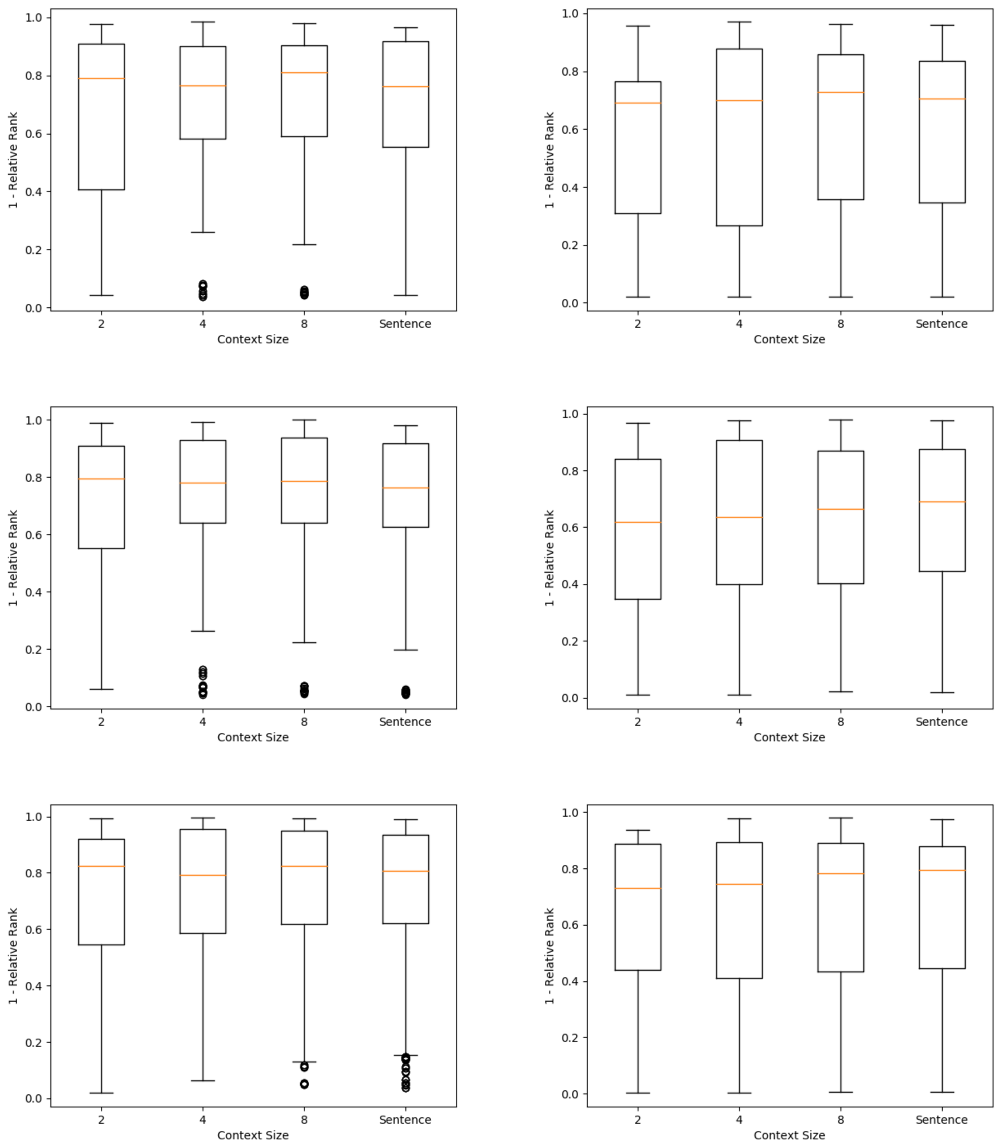


FIGURE 5.20: Influence of Context on German-to-English (left) and English-to-German (right) State Embedding Translations. From top to bottom: Most common 500, 1000 and 2000 words.

The first observation is that the relative ranks are significantly higher than they are in the monolingual evaluation. For small vocabularies, large context windows pose a disadvantage; though, with an increasing number of words and context window size, the relative ranks begin to run asymptotically towards an upper bound, and results at the lower end of the scale are more often outliers than regularities. Also, the difference between word and state embeddings is existent, but shrinks with growing vocabularies and context sizes. This behavior is in stark contrast to the one noticed during the monolingual evaluation (see plots in Figure 5.6 and Figure 5.7). Generally, German-to-English translations show a equal or better quality than their English-to-German counterparts.

In the following chart, the effect of word embedding dimensions is documented.

Again, the left-hand side graphs show German-to-English, the right-hand side ones English-to-German translations, and top, middle, and bottom graphs depict the situation for the most common 500, 1000, and 2000 words.

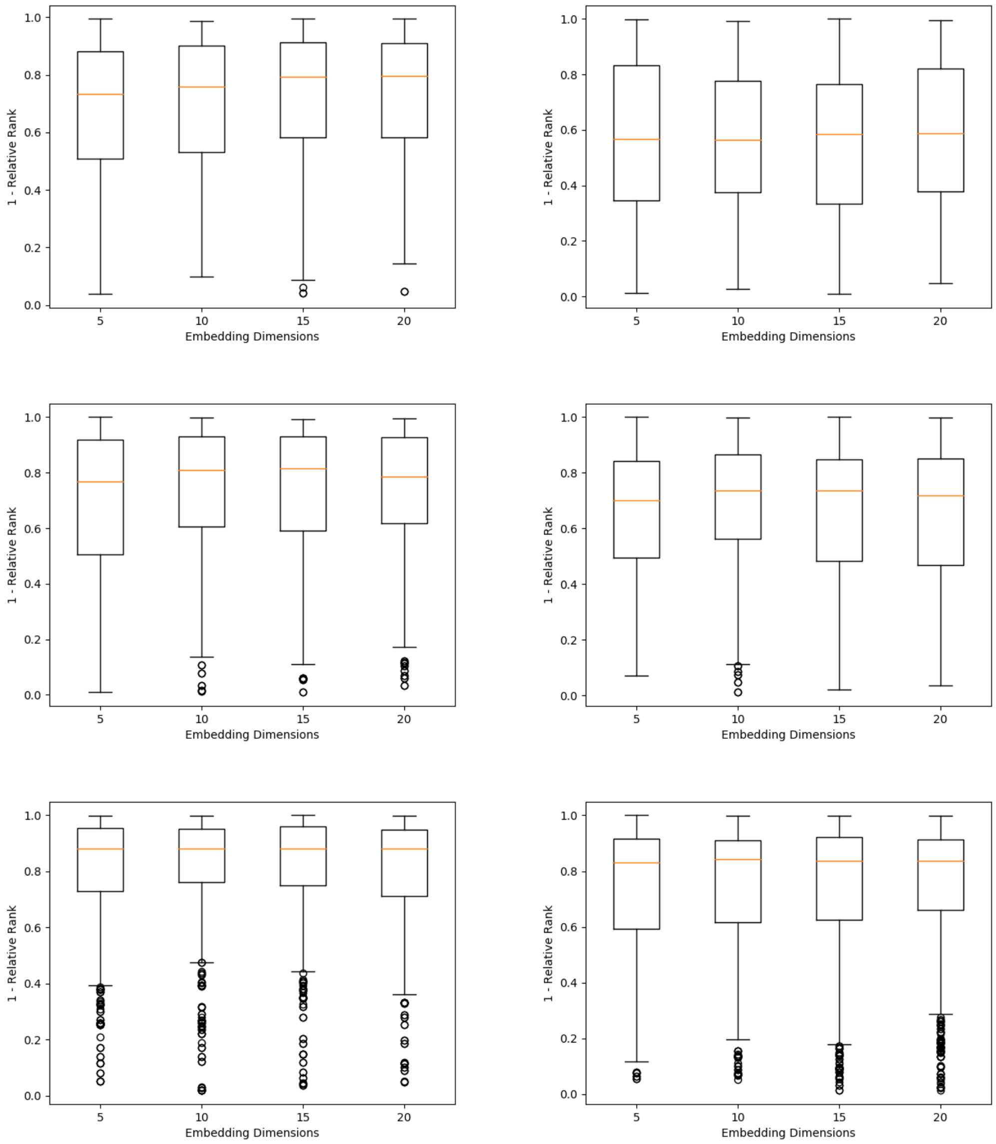


FIGURE 5.21: Influence of Dimensions on German-to-English (left) and English-to-German (right) Word Embedding Translations. From top to bottom: Most common 500, 1000 and 2000 words.

Similarly, the outcomes for state embeddings are plotted:

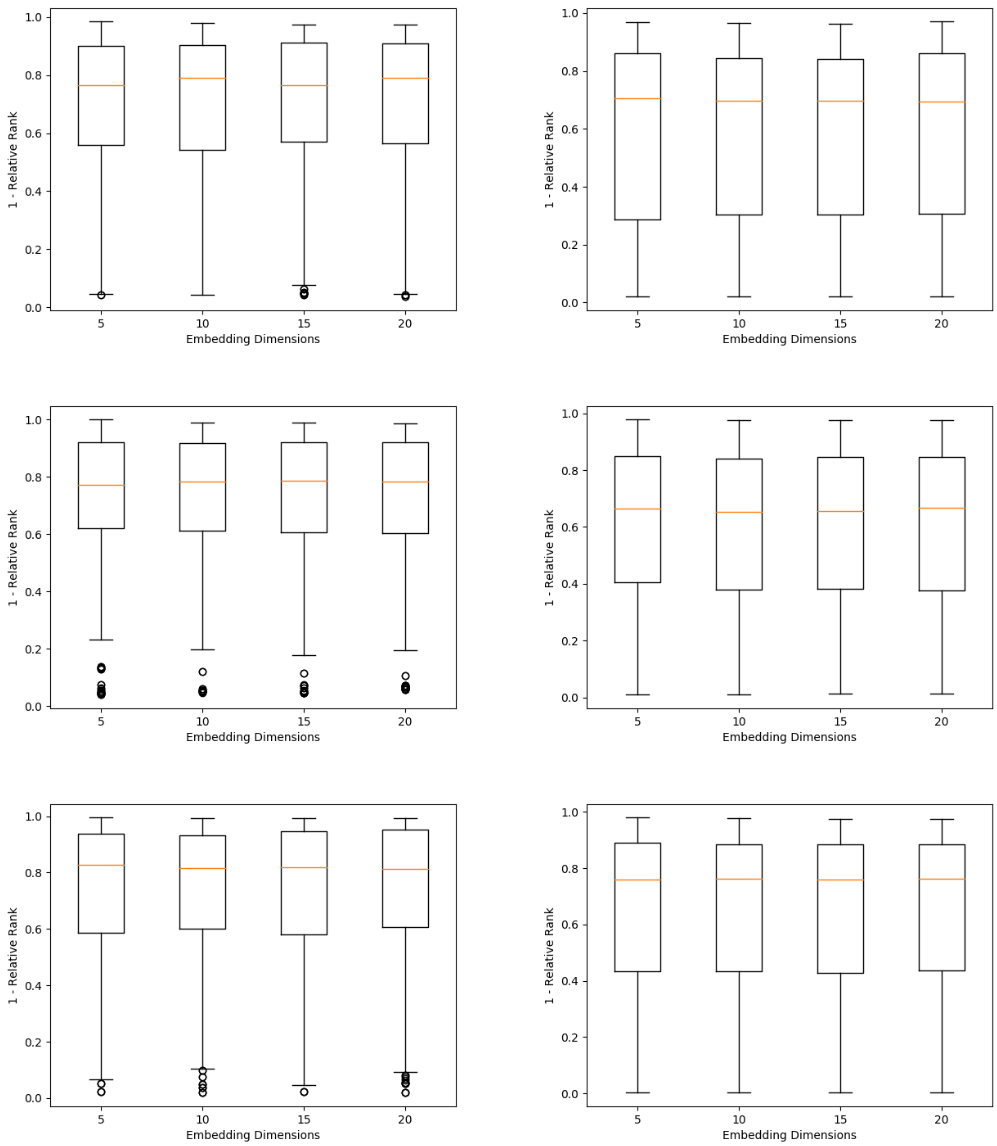


FIGURE 5.22: Influence of Dimensions on German-to-English (left) and English-to-German (right) State Embedding Translations. From top to bottom: Most common 500, 1000 and 2000 words.

The two charts confirm the first impression given by figures 5.19 and 5.20 Interestingly, the embedding size seems to have less impact than the actual vocabulary size, since the medians remain a comparable level within one vocabulary. Furthermore, with a growing number of words, low relative ranks are increasingly becoming outliers. These observations can be maintained for both state and word embeddings. German-to-English translations show again overall a better performance than English-to-German translations. Generally, state embeddings yield slightly lower to almost equal median relative ranks throughout all trials, while performing even better for English-to-German translations among the top 500 words.
Upcoming figures examine the relative ranks for each PoS-tag. As before, plots on

the right show the German-to-English, and ones on the left English-to-German translations; the top/ middle/ bottom row represent the most common 500/ 1000/ 2000 words.

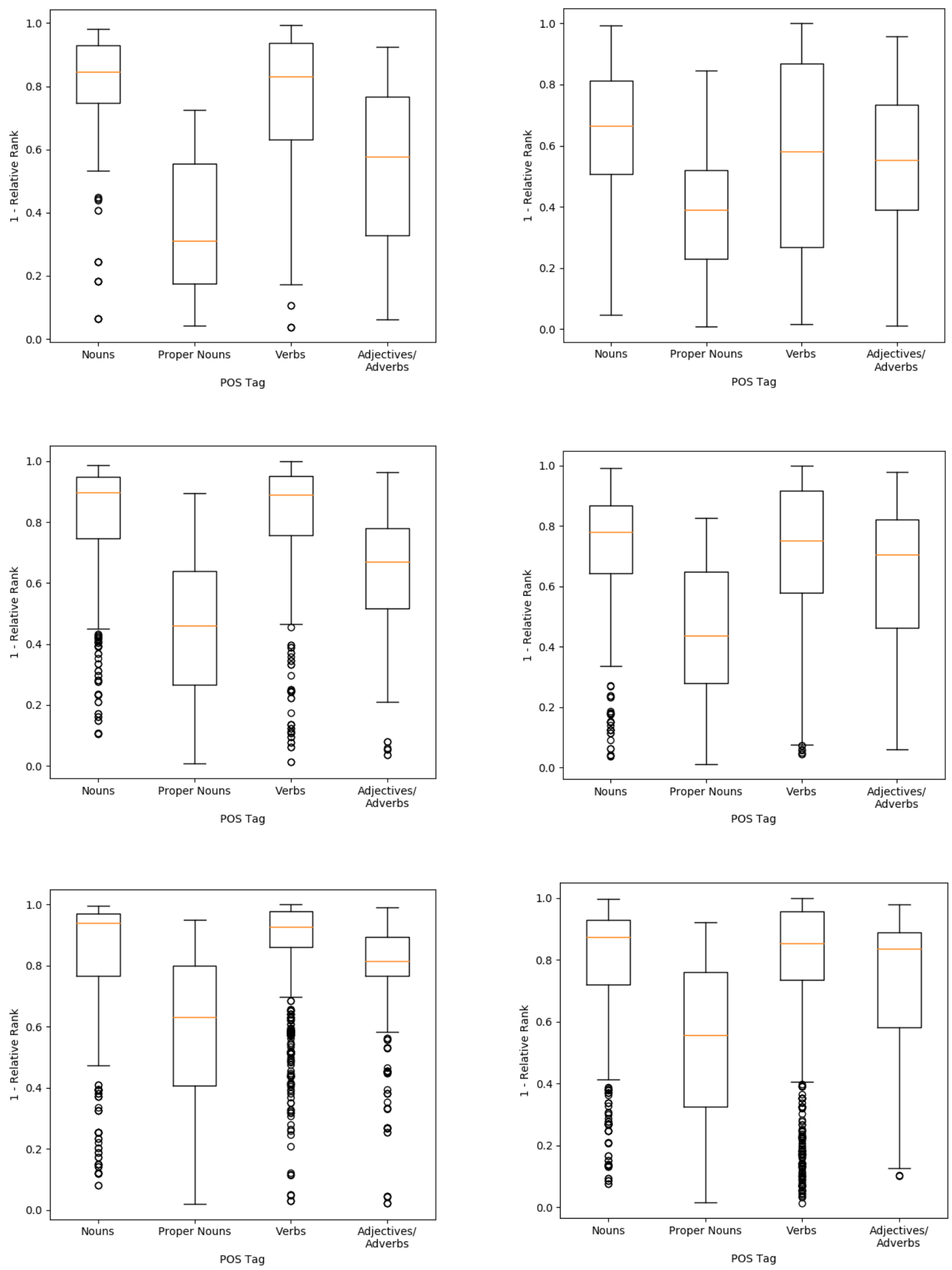


FIGURE 5.23: Comparison of Results for PoS-Tags of German-to-English (left) and English-to-German (right) Word Embedding Translations. From top to bottom: Most common 500, 1000, 2000 Words.

In the same manner, the performance of state embeddings is presented:

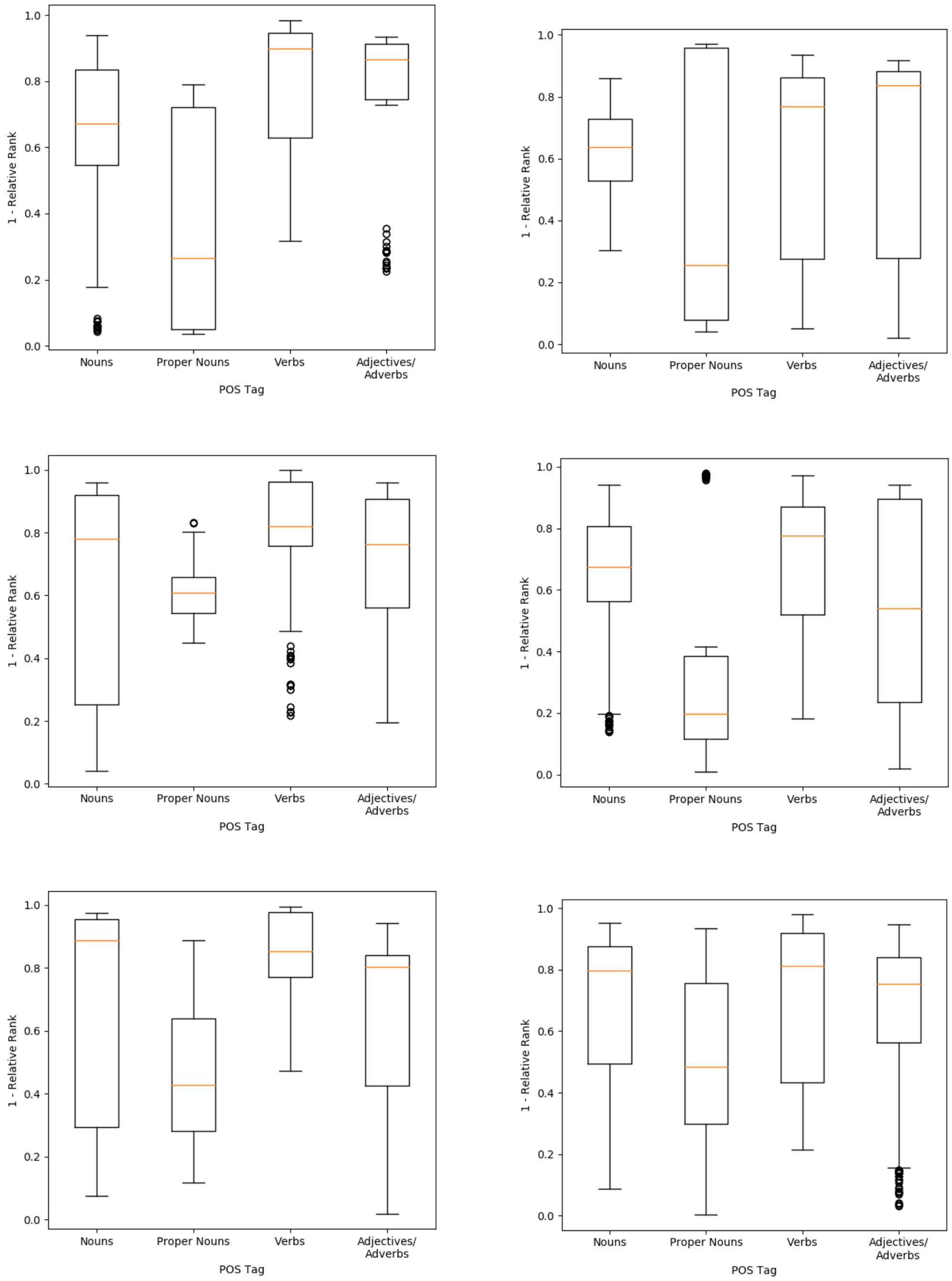


FIGURE 5.24: Comparison of Results for PoS-Tags of German-to-English (left) and English-to-German (right) State Embedding Translations. From top to bottom: Most common 500, 1000, 2000 Words.

Throughout all experiments, nouns have the highest relative ranks, followed by verbs, adjectives/ adverbs, and, significantly behind, proper nouns. As in previous charts, the translational quality improves with larger vocabularies, state embeddings perform slightly worse than word vectors, and German-to-English translations have generally a higher relative rank than English-to-German translations. In addition, there is a trend to wider quartiles for state embeddings, most prominently in the German proper noun translations in the smallest vocabulary level.

Analogously to tables 5.3 and 5.4, the following two overviews present the highest-scoring parameter setups, consisting of context size, embedding mode, and embedding size, of TRANSRANK. First, for German-to-English translations:

Vocabulary Size	Context Size	Embedding Mode	Embedding Size	Average Relative Rank[11]	Average Rank[12]
500	1	word	15	≈0.67543	190
1000	1	state	10	≈0.71241	335
2000	4	word	15	≈0.81880	504

TABLE 5.5: Best on Average Parameter Settings for German-to-English Translations

Secondly, for English-to-German:

Vocabulary Size	Context Size	Embedding Mode	Embedding Size	Average Relative Rank	Average Rank
500	4	State	20	≈0.62058	190
1000	Sentence	Word	5	≈0.66560	335
2000	Sentence	Word	20	≈0.74845	504

TABLE 5.6: Best on-Average Parameter Settings for English-to-German Translations

The best on-average results demonstrate, why one-dimensional box-plots suffice for an initial analysis, especially, when the results are less-than-mediocre, but not for an advanced evaluation. Although box-plots expose certain tendencies within the data for fixed parameters, their interaction cannot be grasped. Although the medians of state embeddings prove to be worse than word embeddings, they appear in both translation directions among the best performing settings; the same applies to embedding dimensionality and context size. Tables 5.5 and 5.6 emphasize that TRANSRANK improves with regards to relative ranks over the given word vectors. The results shown are, in the worst case, around 0.12 relative ranks better than the expected value of the uniform distribution.
Last but not least, the hypothesis on the improvement of translations of lower vocabulary levels is tested.

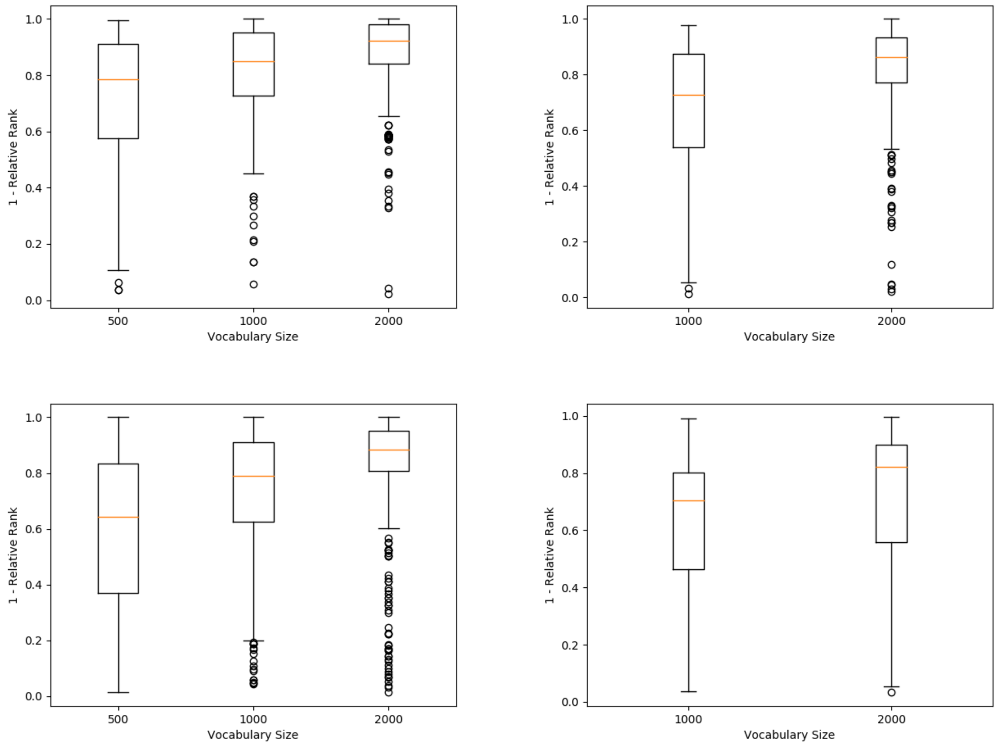


FIGURE 5.25: Differences of Results of German-to-English (left) and English-to-German (right) Translations from lower vocabulary sizes with Word Embeddings. Words from the top 500 (left) and top 1000 words (right).

In the same fashion, the results for state embeddings are arranged:

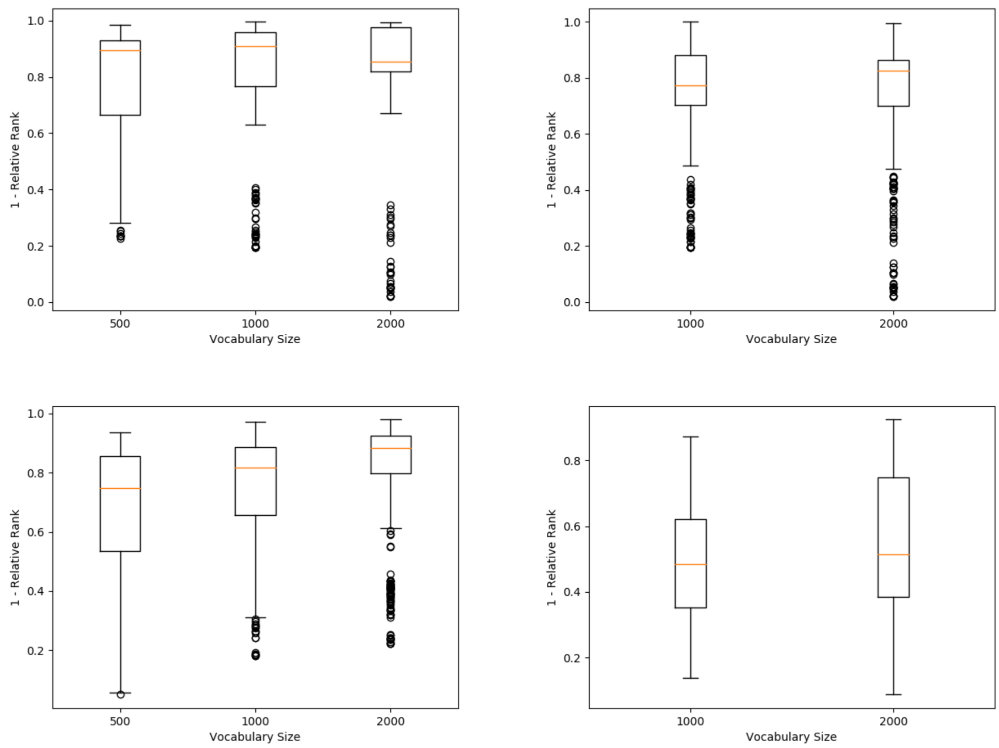


FIGURE 5.26: Differences of Results of German-to-English (left) and English-to-German (right) Translations from lower vocabulary sizes with State Embeddings. Words from the top 500 (left) and top 1000 words (right).

Contrary to how the monolingual evaluation answered this question, for translations, median outcomes for lower vocabularies are always improved with more words. However, this comes at the cost more outliers towards the lower end.
Over the course of the following paragraphs, the evaluation and results of the relates approaches are concisely presented in the order of their appearance in Chapter 4. If not explicitly mentioned, details on data preparation, context- or vector-sizes are not explicitly described by the publications in question.

Canonical Correlation Analysis To evaluate their approach, Haghighi et al. (2008) use four language pairs: English-Arabic (`EN-AR`), English-Chinese (`EN-CH`), English-French (`EN-FR`), and English-Spanish (`EN-ES`). For `EN-AR`, the 1994 Proceedings of the `UN` parallel corpora are used; to mitigate the positive effect of parallel sentences, the English set contains the first, and the Arabic the second 50,000 sentences. `EN-CH` employs the `Xinhua` parallel news corpus, again with the first 50,000 sentences reserved for English, and the second ones for Chinese. The same procedure is applied to `EN-FR`, only with the `Europarl` corpus. Thus, the data of these three language pairs comes from the same domain, only with distinct sentences. In order to document the influence of the corpus the lexica are based on, `EN-ES` is evaluated on three different types of corpora: Once `Europarl` with parallel sentences (indicated by suffix `-P`), and

once in the same fashion as EN-FR (-D), 3851 Wikipedia articles on the same topics (-W), just with non-parallel sentences, and lastly 100,000 sentences from the Gigaword corpus for both English and Spanish, ensuring that also lexica on non-parallel sentences and unrelated domains are created (-G).
The word vectors consist of orthographic and contextual information. Each substring of length below or equal to three, as well as nouns within a context window of four are included as (presumably one-hot) features. Only noun *types* (meaning, lemmatized tokens) are considered in the monolingual lexica.
Evaluation dictionaries for EN-CH, EN-FR, and EN-ES are implemented with the Wictionary online dictionary; in the case of EN-AR, an alignment model is applied on 100,000 parallel sentences from the UN parallel corpora to extract source-target-pairs. Seed lexica, if not marked otherwise, are of size 100, selected from the top 2000 most common nouns, and included as connections in the translation graph. The other approach consists of inducing the seed lexicon by edit distance. Table 5.7 gives a first overview on the performance of CCA:

Setting	$p_{0.1}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	Best-F_1
EDITDIST	58.6	62.6	61.1	—	47.4
ORTHO	76.0	81.3	80.1	52.3	55.0
CONTEXT	**91.1**	81.3	80.2	65.3	58.0
MCCA	87.2	**89.7**	**89.0**	**89.7**	**72.0**

TABLE 5.7: Results for EN-ES-W

Above, the results for EN-ES-W are shown; EditDist refers to the standard string distance, Ortho and Context to the orthographic and contextual features, and MCAA denotes the best-performing feature set. Columns starting with p_x give the precision at a certain recall x; best-F1 stands for the best F1 score obtained "over all possible thresholds and various precisions". This is possible, because the actual translations are retrieved from the weighted bipartite graph. Depending on thresholds, the precision - or recall, respectively - changes. It is worth noting that Haghighi et al. (2008) do not penalize the precision metric, if a proposed translation does not exist in the evaluation lexicon; also recall is not sanctioned for not retrieving all possible translations. The next figure shows the effect of the corpus:

Setting	$p_{0.1}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	Best-F_1
EN-ES-G	75.0	71.2	68.3	—	49.0
EN-ES-W	87.2	89.7	89.0	89.7	72.0
EN-ES-D	91.4	94.3	92.3	89.7	63.7
EN-ES-P	97.3	94.8	93.8	92.9	77.0

TABLE 5.8: Effect of the Corpus on Precision/ F1

As expected, the precision increases, the closer the domain and sentences are, on which the monolingual lexica are built. The table below investigates to which extent bilingual seed lexica account for the translation quality:

Corpus	$p_{0.1}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	Best-F_1
EDITDIST	58.6	62.6	61.1	—	47.4
MCCA	91.4	94.3	92.3	89.7	63.7
MCCA-AUTO	91.2	90.5	91.8	77.5	61.7

TABLE 5.9: Effect of the Seed Lexica on Precision/ F1

It can be seen that a 'correct' seed lexicon outperforms the edit distance, even for a comparatively similar language pair as English and Spanish. The last figure gives an overview on the other pairs English-Arabic, English-Chinese, and English-French:

Languages	$p_{0.1}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	Best-F_1
EN-ES	91.4	94.3	92.3	89.7	63.7
EN-FR	94.5	89.1	88.3	78.6	61.9
EN-CH	60.1	39.3	26.8	—-	30.8
EN-AR	70.0	50.0	31.1	—-	33.1

TABLE 5.10: Differences in Precision/ F1 for other Language Pairs

With growing orthographic distance and larger differences in morphology, the translation quality degrades. Overall, it is also evident that a increasing recall (meaning, the system finds more of the correct translation pairs), precision decreases (i.e., more possible, however wrong translations are identified).

Generative Adversarial Nets Conneau et al. (2017) align 300-dimensional FASTTEXT embeddings trained on `Wikipedia` with a GAN approach. Words which appear less than five times are discarded. Since the quality of embeddings increases with their frequency, the discriminator in the GAN is fed with the 50,000 most frequent words. Translation quality is evaluated for English-Spanish (`en-es`/ `es-en`), English-French (`en-fr`/ `fr-en`), English-German (`en-de`/ `de-en`), English-Russian (`en-ru`/ `ru-en`) and English-Chinese (`en-zh`/ `zh-en`) with gold-standard dictionaries "using an internal translation tool" (Conneau et al., 2017) containing 100,000 word pairs, where one word is mapped not only to one, but multiple possible translations. In each trial, 1500 terms are translated, while setting the set of acceptable targets to 200,000 words. The table below shows the results measured in precision:

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	en-eo	eo-en
Methods with cross-lingual supervision and fastText embeddings												
Procrustes - NN	77.4	77.3	74.9	76.1	68.4	67.7	47.0	58.2	40.6	30.2	22.1	20.4
Procrustes - ISF	81.1	82.6	81.1	81.3	71.1	71.5	49.5	63.8	35.7	**37.5**	29.0	27.9
Procrustes - CSLS	81.4	82.9	81.1	**82.4**	73.5	**72.4**	**51.7**	**63.7**	**42.7**	36.7	**29.3**	25.3
Methods without cross-lingual supervision and fastText embeddings												
Adv - NN	69.8	71.3	70.4	61.9	63.1	59.6	29.1	41.5	18.5	22.3	13.5	12.1
Adv - CSLS	75.7	79.7	77.8	71.2	70.1	66.4	37.2	48.1	23.4	28.3	18.6	16.6
Adv - Refine - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9	20.7	20.6
Adv - Refine - CSLS	**81.7**	**83.3**	**82.3**	82.1	**74.0**	72.2	44.0	59.1	32.5	31.4	28.2	**25.6**

TABLE 5.11: Results by Conneau et al. (2017)

Procruste stands for the supervised baseline using only the solution to Procruste's Problem (as described by Artetxe et al. (2017)), with a seed dictionary of 5000 word pairs. By ADV, the unsupervised, adversarial approach is meant. Suffixes behind the dash denote nearest neighbour (-NN), inverted softmax (-ISF), which are both suggested by related work to find the most suitable translation, and cross-domain similarity local scaling (-CSLS), as suggested by Conneau et al. (2017). The final refinement using Procruste's solution on the obtained translation matrix is abbreviated by -Refine-.
During all experiments, the CSLS variable K is set to 10, as all tests show that results are relatively stable for K=*5*, K=*10*, and K=*50*. A detailed summary of all training parameters can be found in (Conneau et al., 2017)
Numbers in Table 5.11 emphasizes the strength of Conneau et al's approach - precisions for unsupervised methods differ at most 7.7% (in case of `en-ru`) from the supervised ones, and even then, their proposed CSLS translation process is superior to NN and ISF (apart from `zh-en`). This proves that unsupervised approaches can compete with supervised counterparts. The next table shows the numbers in comparison to related work:

	English-to-Italian			Italian-to-English		
	P@1	P@5	P@10	P@1	P@5	P@10
Supervised - `WaCky`						
(Mikolov et al., 2013)	38.8	48.3	53.9	24.9	41.0	47.4
(Artetxe et al., 2017)[13]	39.7	54.7	60.5	33.8	52.4	63.6
Procrustes-CSLS	44.9	61.8	66.6	38.5	57.2	63.0
Unsupervised - `WaCky`						
Adv-Refine-CSLS	45.1	60.7	65.1	38.3	57.8	62.8
Supervised - `Wikipedia`						
Procrustes-CSLS	63.7	78.6	81.1	56.3	76.2	80.6
Unsupervised - `Wikipedia`						
Adv-Refine-CSLS	**66.2**	**80.4**	**83.4 58.7**	**76.5**	**80.9**	

TABLE 5.12: Comparison of English-Italian Translation Accuracies for Embeddings trained on `WaCky` and `Wikipedia`

The table above shows that the results obtained by Conneau et al. (2017) are not only on their own, but also comparatively very good. In relation to Mikolov et al. (2013) and Artetxe et al. (2017), who both use CBOW vectors, accuracy on the first, top five, and top ten translations is always better. This indicates a clear benefit of incorporating sub-word information. It furthermore shows that FASTTEXT embeddings trained on `Wikipedia` yield a significantly higher performance than those trained on `WaCky`. According to Conneau et al. (2017), this is due to the `Wikipedias` more similar co-occurrence statistics. Together with the adversarial approach, which already gives reasonable results, even on distant language pairs, the refinement step induces an additional gain. That is, because after adversarial learning, the translation matrix provides more acquired training instances, than the baseline supervised seed dictionary contains.
Besides translation, Conneau et al. (2017) also investigate cross-lingual word similarity and sentence retrieval; however, as this thesis is for starters only concerned with unsupervised translation, the interested reader is referred to their paper.

Neural Network Optimization Mikolov et al. (2013) evaluate their NN-based method on Czech, English, Spanish, and Vietnamese. Word vectors for the former three are built on the `WMT11` data set, consisting mainly of the `Europarl` and additionally the `News Commentary` corpus, while for the latter, the `Google News` data is used. To prove scalability, English and Spanish word vectors are later also computed on the larger `Google News` corpora with "several billion words"(Mikolov et al., 2013). Data preparation involves tokenization, removal of duplicate sentences, named entities, and punctuation in general, and substitution of written numeric values by digits. Additionally, short term phrases are subsumed under an extra string, if words are more likely to co-occur in certain contexts than their unigram frequency suggests, following the strategy from Mikolov et al. (2013a). For instance, in the case of *ice cream*, *ice*, *cream*, and *icecream* would be added to the vocabulary. Gold standard translations are retrieved from GOOGLE TRANSLATE for the 5000 most common terms in each language, which serve as seed lexicon. As test set, words of frequency ranks between 5000 and 6000 are used. The following table shows the size of training tokens and vocabulary sizes, which contain words occurring at least five times in the corpus:

Language	**Training tokens**	**Vocabulary size**
English	575M	127K
Spanish	84M	107K
Czech	155M	505K

TABLE 5.13: Overview on Training-Set and Vocabulary Sizes

In addition, the Vietnamese training set consists of 1.3 billion phrases, which are equivalent to English words and short phrases. In order to assess their method properly, Mikolov et al. (2013) provide two baselines: Once, the edit distance between goal and all possible target words is calculated, and secondly, more elaborated, a count-based approach, where for each language, a co-occurrence matrix comprising all dictionary terms is created. (Row) Vector entries are log- and ℓ_2-length normalized. Using the gold standard translations, every term is then mapped to its bilingual counterparts, and afterwards, for each test word in the source language, the closest target term regarding cosine distance (following the standard translation procedure) is selected. Table 5.14 presents the accuracies for English-to-Czech/ English-to-Spanish translations (and vice versa):

Translation	**Edit Distance**		**Word Co-occurrence**		**Translation Matrix**		**ED + TM**		**Coverage**
	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5	
En → Sp	13%	24%	19%	30%	33%	51%	43%	60%	92.9%
Sp → En	18%	27%	20%	30%	35%	52%	44%	62%	92.9%
En → Cz	5%	9%	9%	17%	27%	47%	29%	50%	90.5%
Cz → En	7%	11%	11%	20%	23%	42%	25%	45%	90.5%

TABLE 5.14: Accuracies tor English-Czech/ English-Spanish Translations

Analysis is split between two measurements: P@1, where only the closest target word is taken as translation, and P@5, which uses the top 5 closest target terms for evaluation. ED + TM, i.e. edit distance plus translation matrix, denotes "a weighted combination of similarity scores given by both techniques"(Mikolov et al., 2013). By coverage, the percentage of targets produced by GOOGLE TRANSLATE is meant, which are also part of the collected vocabularies. It is worth noting that vector sizes do not necessarily need to correspond; for example, English to Spanish translations have the highest accuracy for 800-dimensional English and 200-dimensional Spanish word vectors. Edit distance performs better for English and Spanish, than English and Czech, since those two are more distant. Throughout the experiments, results for the translation matrix are reasonable; when combined with edit distance, baselines are more than doubled. As can be seen from the edit distance baseline, the improvement between the plain translation matrix and its combination with edit distance is much larger for English-Spanish than English-Czech translations (and vice versa). The next two tables explore the scalability of the approach. Therefore, `Google News` corpora in English and Spanish are employed. Table 5.15 presents the connection between cosine thresholds, accuracies, and vocabulary coverage for the original translation matrix:

Threshold	Coverage	P@1	P@5
0.0	92.5%	53%	75%
0.5	78.4%	59%	82%
0.6	54.0%	71%	90%
0.7	17.0%	78%	91%

TABLE 5.15: Accuracies of English-Spanish Translation Matrix with Cosine Thresholds

Results for the translation matrix combined with edit distance are given below:

Threshold	Coverage	P@1	P@5
0.0	92.5%	58%	77%
0.4	77.6%	66%	84%
0.5	55.0%	75%	91%
0.6	25.3%	85%	93%

TABLE 5.16: Accuracies of English-Spanish Translation Matrix combined with Edit Distance and Cosine Thresholds

Unsurprisingly, a higher threshold leads to a more reliable translations. On the downside, the coverage decreases, as a much lower percentage of words is matched to bilingual counterparts. However, this subset of trustworthy source-target pairs can be used to establish a new seed dictionary, or to discard definite incorrect translations. The following two graphs show the gain and decline in precision when the number of training words increases (left) and the test words become more infrequent (right); both for `English-to-Spanish` translation.

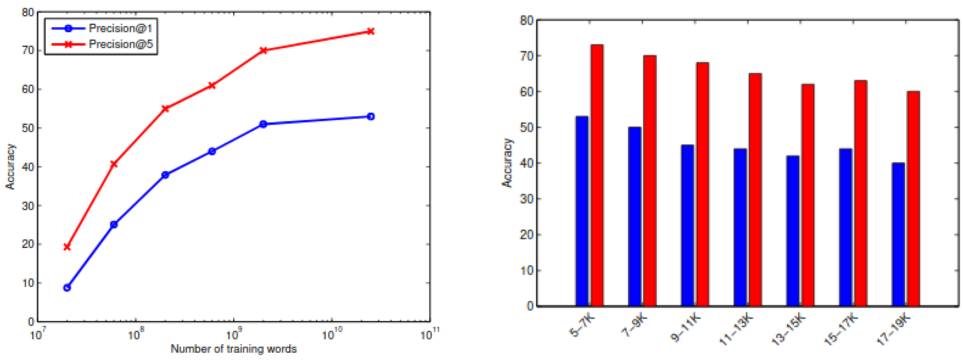


FIGURE 5.27: Precision of English-to-Spanish Translations for increasing Training-Set Size (left) and increasingly infrequent Words (Right)

The plot on the left supports the aforementioned hypothesis that with a growing vocabulary size, resolution of meaning over more context is facilitated, whereas the graph on the right-hand side simply demonstrates that infrequent words occur less often in contexts and are thus harder to translate.

Last but not least, English and Vietnamese translations are evaluated, to exemplify translation between two very distant languages:

Threshold	Coverage	P@1	P@5
En $\rightarrow$ Vn	87.8%	10%	30%
Vn $\rightarrow$ En	87.8%	24%	40%

TABLE 5.17: Accuracy for English-Vietnamese Translations

Here, edit distance is not combined with the translation matrix, because "the concept of a word is different than in English"(Mikolov et al., 2013). This, plus the fact that there is a large number of synonyms in the Vietnamese training set, leads to comparatively lower accuracies, especially for English-to-Vietnamese translations.

Procruste's Problem As basis, Artetxe et al. (2017) train CBOW word vectors in English, Finnish, German and Italian. The context window is set to five, embedding dimension to 300, subsampling threshold to $1e^{-5}$, and the number of negative samples to ten. English vectors are trained on a 2.8 billion words combination of `ukWaC`, `Wikipedia`, and `BNC` (British National Corpus), Italian vectors on 1.6 billion words from `itWaC`, German ones on the 0.9 billion words corpus `SdeWaC`, and Finnish word vectors on 2.8 billion words from `Common Crawl`[14]. The $\star$`WaC`$\star$ corpora are described in detail in (Baroni et al., 2009) and consist of cleaned, linguistically preprocessed web crawls. Each monolingual vocabulary is restricted to its 200,000 most common words. Pre-informed seed dictionaries comprise of 25, 50, 75, 100, 250, 500, 1000, 2500, and 5000 randomly sampled entries, being derived from 5000 most frequent `Europarl` word alignments. 1500 word pairs, also from `Europarl` word alignments, which are uniformly distributed over five frequency bins, build the held-out test set. Besides, Artetxe et al. (2017) also implement an unsupervised *numeral* dictionary which consists of common strings in the monolingual vocabularies matching the regular expression $[0-9]^+$. This gives an initial dictionary of 2772 numerals for English-Italian, 2148 for English-German, and 2345 for English-Finnish. The process of determining the dictionary matrix $\mathbf{D}$, is aborted, when the average of differences in all updates in

$$\mathbf{D}[i][j] = \begin{cases} 1, \text{ if } \mathbf{x}_i\mathbf{W}\mathbf{y}_j^T \text{ is maximal} \\ 0 \text{ otherwise.} \end{cases} \tag{5.14}$$

has reached $1e^{-6}$. The authors note that this takes "usually" less than 100 iterations. Artetxe et al. (2017) also evaluate cross-lingual word similarity, which is not discussed here. Table 5.18 shows an excerpt of translation accuracies obtained by Artetxe et al. (2017), compared to Mikolov et al. (2013) on the same data sets:

[14] `http://statmt.org/wmt16/translation-task.html` [Accessed: 6.8.2020]

Publication	English-Italian			English-German			English-Finnish		
	5000	25	num.	5000	25	num.	5000	25	num.
(Mikolov et al., 2013)	34.93	0.00	0.00	35.00	0.00	0.07	25.91	0.00	0.00
(Artetxe et al., 2017)	39.67	37.27	39.40	40.97	39.60	40.27	28.72	28.16	26.47

TABLE 5.18: Accuracies for the Approaches of Mikolov et al. (2013) and Artetxe et al. (2017). Column Numbers refer to the seed dictionary size, *num.* to the unsupervised numerical Dictionary.

Their proposed method performs much better than the on by Mikolov et al. (2013), especially for small seed dictionaries. The unsupervised numerical dictionary is on par with the large 5000 seed dictionary. Given the mere size of those dictionaries (ranging from 2148 for English-German to 2772 for English-Italian) this not surprising. However, as in the case of English-Finnish translations, the difference is larger. This might be due to the smaller Finnish corpus, which could penalize the performance for the rather infrequent numbers, compared to the 5000 most common words in the training seed dictionary. Regarding a more detailed analysis, roughly one third of all errors are due to morphological variants of the target word (Artetxe et al., 2017). Another error source stems from mis-aligned named entities (e.g., *Volvo* instead of *BMW*), which account for a third of the remaining erroneous translations. In most of the other cases, the words are either strongly related (via synonymy, or a similar semantic field) or rather metaphorically. Partial translations of multi-words are also a problem (cf. structural mismatches, idioms and collocations in Chapter 2). Furthermore, Artetxe et al. (2017) observe that sometimes a rare word appears repeatedly among translations; an issue which is familiar from the analysis of the proposed word vector methodology in Section 5.1.2.1.

To take a closer look on the connection between seed dictionary size, number of iterations and accuracy, Artetxe et al. (2017) provide the following graph for English-Italian translations:

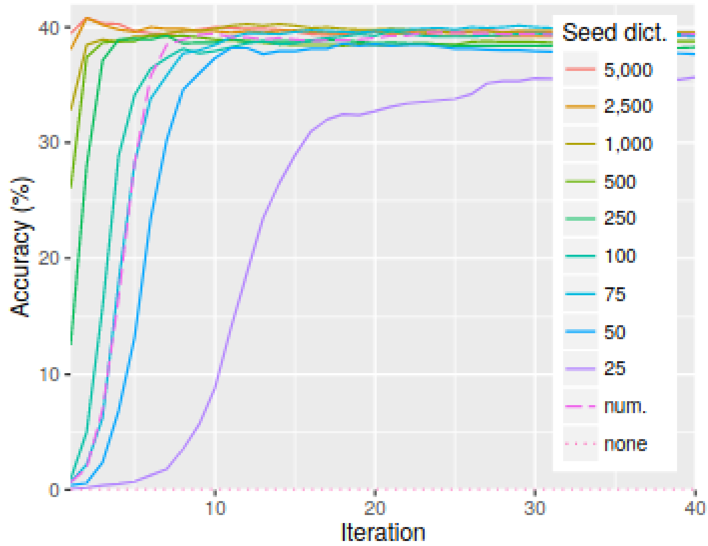


FIGURE 5.28: Development of Translation Quality for English-Italian over Time with different Seed Dictionaries.

Without a seed dictionary, there is no gain in performance at all. However, if its size increases, the 'learning' curve becomes steeper, and with a sufficient number of iterations, accuracies converge. Artetxe et al. (2017) note that the algorithms' reliance on previously calculated alignments stored in **D** prevent from learning "degenerated" patterns on a large scale; meaning, "it is guaranteed that the value of the optimization objective will improve (or at least remain the same)." This reasoning is empirically supported by Figure 5.28 above: Independently from the initial seed dictionary, translation quality improves, and converges as well. Starting with a totally random initialization, though, does not suffice, because then the process "tends to get stuck in poor local optima".

SIMRANK In their SIMRANK algorithm, Dorow et al. (2009) employ verb-object relations between verbs and nouns compiled from the 100 million words `BNC` and the `Huge German Corpus` comprising of 180 million words of newspaper text. Verbs and nouns are arranged in a bipartite graph, such that verbs are not connected among themselves, and neither nouns. While other syntactical relations would be possible to add or test, too, Dorow et al. (2009) hypothesize that verb-object connections perform best in disambiguating contexts. Single nodes, which have only one neighbour, are excluded, as those do not contribute relevant meaning, as well as relationships which occurr less than three times in the corpora. All words are lemmatized and filtered through stop word lists; furthermore, English compounds are substituted by their heads, and English verbs are extended by prepositions (for instance, *put* + *off*). English and In the case of English, their graph contains after pruning 4,926 nodes, distributed over 3,365 nouns and 1,561 verbs, connected via 43,762 links. In German, there is a total of 3,074 nodes, which comprise of 2,207 nouns and 867 verbs,

with overall 15,386 links. To measure correspondences between English and German, (damping) constant c is set to 0.8, as suggested by Jeh and Widom (2002) and Brin and Page (1998), and the number of iterations is fixed to six.
The seed dictionary contains reference translations from the online dictionary dict.cc[15] for all words except held-out test sets. These test sets include each 50 nouns and verbs in English and German for three frequency levels (>100, 20-100, and ≤20 occurrences). As laid out in section Section 4.3.1, the similarity matrices between English and German verbs and nouns contain the transition probabilities of two verbs or nouns in question. Thus, the closest translation for a given test word is the one having the largest entry in its row vector. Dorow et al. (2009) use for their analysis the relative rank; however, they do not subtract it from one, meaning, a value of zero denotes the closest, and an outcome of about one the farthest translation.
Table 5.19 gives the mean of results for English-to-German/ German-to-English translations, divided into verbs and nouns, and their frequency level.

English						German					
Low		Mid		High		Low		Mid		High	
N	V	N	V	N	V	N	V	N	V	N	V
0.313	0.228	0.253	0.288	0.253	0.255	0.232	0.247	0.205	0.237	0.211	0.205

TABLE 5.19: Mean relative Ranks for English-to-German/ German-to-English Translation

Graph Figure 5.29 shows the frequency distribution of relative ranks:

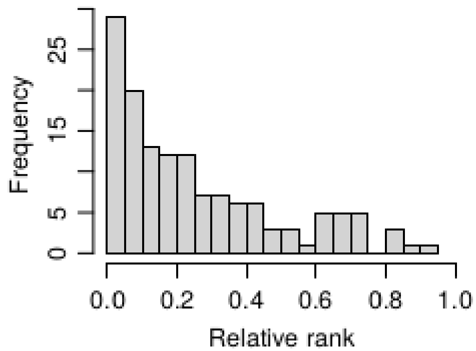


FIGURE 5.29: Distribution of relative Rank Frequencies of Reference Translations

It can be seen that mean relative ranks for frequent words are higher than for infrequent ones. Differences between verbs and nouns are irregular, however, a large gap between translations for infrequent English nouns and verbs is notable. Also, results for English-to-German translations are slightly worse than vice versa. Overall, the results are reasonable though, as indicated by the plot above: SIMRANK often ranks correct translations high, and very seldomly low.

[15] http://www.dict.cc/ [Accessed: 6.8.2020]

But what about the similarity between the top-ranked translations? Table 5.20 presents exemplarily the top translations for two English and German words:

Test word	Top 10 predicted translations	Ranks
sanction	Ausgangssperre Wirtschaftssanktion Ausnahmezustand Embargo Moratorium *Sanktion* Todesurteil Geldstrafe Bußgeld Anmeldung	Sanktion(6) Maßnahme(1407)
delay	anfechten revidieren zurückstellen füllen verkünden quittieren vertagen verschieben aufheben respektieren	verzögern(78) aufhalten(712)
Kosten	hallmark trouser blouse makup uniform armour robe testimony witness jumper	cost(285)
öffnen	unlock lock usher step peer shut guard hurry slam close	open(12) undo(481)

TABLE 5.20: Overview over Test Words and their determined/ actual Translations

Translations are *Ausgangssperre - lockdown/curfew*, *Wirtschaftssanktion - economic sanctions/ embargo*, *Ausnahmezustand - (state of) emergency*, *Embargo - embargo*, *Moratorium - moratorium*, *Sanktion - sanction*, *Todesurteil - death penalty*, *Geldstrafe - fine/ penalty fee*, *Bußgeld - fine/ penalty fee*, *Anmeldung - application/ registration*, *anfechten - (to) challenge revidieren - (to) revise*, *zurückstellen - (to) defer/ (to) postpone*, *füllen - (to) fill*, *verkünden - (to) announce*, *quittieren - (to) quit/ (to) confirm*, *vertagen - (to) adjourn*, *verschieben - (to) postpone*, *aufheben - (to) cancel/ (to) lift*, *respektieren - (to) respect*. Except for *Kosten*, at least the closest targets determined by the algorithm belong to the same semantic field, and relate to the source word. Dorow et al. (2009) explain the answers given for *Kosten* by its co-occurrence with the ambiguous word *tragen* ((to) wear/ (to) bear). The opposite results for *öffnen* (*(to) close*, *(to) lock*, *(to) shut*) emphasize the limitations of only including verb-noun relations.

COSIMRANK Following Dorow et al. (2009), Rothe and Schütze (2014) employ linguistically informed edges; besides verb-object, they also consider adjective-noun and noun-noun relations. Based on the English and German graphs compiled by Laws et al. (2010) with lemmatized, tagged and parsed `Wikipedia` articles, the English graph contains 40,002 vertices, while the German network consists of 47,439 nodes.

Table 5.21 gives an overview on the distribution of vertices and edges for both languages:

Graph statistics

nodes		nouns	adjectives	verbs
	de	34,544	10,067	2,828
	en	22,258	12,878	4,866
edges		ncrd	amod	dobj
	de	65,299	417,151	143,905
	en	288,878	686,069	510,351

TABLE 5.21: Overview of Nodes and typed Edges

Adjective-noun edges are removed if they occur less than three times. For noun-noun pairs, the same applies, in addition to a filter which deletes nouns if their count is below 100. Verb-object relationships remain unaffected, as they pose the smallest data set. Furthermore, pairs with low association scores are removed (see Laws et al. (2010) for details). Organized in typed adjacency matrices, edges are frequency-weighted and normalized, meaning, the edge weight corresponds to the transition probability.
Examples for edge types are shown in the oversight below:

Edge types

relation	entities	description	example
amod	a, v	adjective-noun	a fast car
dobj	v, n	verb-object	drive a car
ncrd	n, n	noun-noun	cars and busses

TABLE 5.22: Edge Types between Entities with Examples

Two tasks are evaluated; synonym extraction on the English graph, and lexicon induction. In case of the former, the synonymy test set (referred to as TS68) by Minkov and Cohen (2012) is used, containing 68 words (22 nouns, 22 verbs, and 24 adjectives), each listed with a single appropriate synonym. For the latter, a test set comprising the 1000 most common words from `Wikipedia` (named TS1000), including 660 nouns, 200 verbs, and 140 adjectives, is employed. The seed dictionary contains 12,630 word pairs, which is used to instantiate correspondences between bilingual nodes.
Table 5.23 lists the results produced by (typed) COSIMRANK, together with SIMRANK and the cosine-compared personalized PAGERANK (PPR) vectors as baseline. P@1 and P@10 stand for the precision that the correct answer is either the first or among the first ten closest results, and MRR denotes the mean reciprocal rank. In an additional *extended* experiment, the authors analyze the answers given by the system with the help of three native English speakers; if they all agree on one answer being a synonym in at least one meaning, this answer is evaluated as correct.

	P@1	P@10	MRR
	one-synonym		
PPR+cos	20.6%	52.9%	0.32
SimRank	**25.0%**	61.8%	**0.37**
CoSimRank	**25.0%**	61.8%	**0.37**
Typed CoSimRank	23.5%	**63.2%**	**0.37**
	extended		
PPR+cos	32.6%	73.5%	0.48
SimRank	**45.6%**	**83.8%**	**0.59**
CoSimRank	**45.6%**	**83.8%**	**0.59**
Typed CoSimRank	44.1%	**83.8%**	**0.59**

TABLE 5.23: Results for Synonym Extraction. The best Result per Column is boldfaced.

Exemplary answers are shown below:

keyword	expected	extracted
movie	film	**film**
modern	contemporary	**contemporary**
demonstrate	protest	**show**
attractive	appealing	beautiful
economic	profitable	financial
close	shut	open

TABLE 5.24: Keywords for Similarity Task with expected and extracted Outcomes.

During all experiments, the decay/ damping factor is set to 0.8, as suggested by previous authors (cf. (Dorow et al., 2009), (Jeh and Widom, 2002), (Brin and Page, 1998)). Every method, besides PPR, which uses 20, is calculated with five iterations. The results reveal that COSIMRANK is equal or better than PPR and SIMRANK baselines. Especially, the distributions of correct ranks for SIMRANK and COSIMRANK have the same mean. In summary, COSIMRANK performs reasonable on monolingual synonym extraction, which is an important criterion for finding similar bilingual terms.

This leads to the second task, translation. Before a test word is translated, it is removed from the seed dictionary, such that its translation is induced from the remaining word pairs. Outcomes for English-to-German translations are given in Table 5.25: COSIMRANK as well as typed COSIMRANK outperform SIMRANK. The differences between both COSIMRANK approaches are negligible. Results for PPR are far behind, and statistically significant worse than those of COSIMRANK. A reason for this behaviour might be that the seed dictionary covers only about one fourth of the vocabulary (12,630 seed dictionary entries vs. 47,439 German words), so, only every fourth vector is used during similarity calculation in PPR. While this is also the

	P@1	P@10
PPR+cos	14.8%	45.7%
CoSimRank	61.1%	**84.0%**
Typed CoSimRank	**61.4%**	83.9%

TABLE 5.25: Results for English-to-German Translation. The best Result per Column is boldfaced.

case for COSIMRANK, it seems to be more stable, as it compares more than one vector (Rothe and Schütze, 2014). Although almost two third of translations are correct, those come at high stakes, as lemmatization, tagging, parsing, and implementing a seed lexicon with more than 10,000 entries include a lot of (human) supervision in programming and data preparation.

These sections presented the outcomes of the advocated methods and compared those to related work. The next chapter discusses the findings, and gives ideas for future work.

Chapter 6

Discussion & Future Work

"The great tragedy of Science - the slaying of a beautiful hypothesis by an ugly fact."
Thomas Henry Huxley, 1870 (Huxley, 1870)

6.1 Discussion

As the last chapter shows, the results are mixed: While the evaluation of the word vectors exhibits that finite-state embeddings are unable to incorporate useful meaning of words passing through, and that standard word embeddings do also not retain information as expected, the translation model performs surprisingly well, given the poor data it operates on. In this section, these advantages and downsides shall be discussed, as well as what could be learned from related methods.

6.1.1 Analysis of the Evaluation

The first question is, why especially state embeddings, but also word embeddings, provide such low relative ranks for the correct answers. After all, in every parameter setting, in ≥50% of more than half of the vocabulary is preferred over the correct answer. Table 6.1 below takes a look on exemplary words, and their relevant states from the automaton:

Word	State Vector	Word	State Vector
high	$\begin{pmatrix}1876 & 7 & 797 & 822 & 1950\end{pmatrix}$	large	$\begin{pmatrix}1876 & 11 & 963 & 971 & 1970\end{pmatrix}$
higher	$\begin{pmatrix}1876 & 7 & 797 & 822 & 1950 & 827 & 829\end{pmatrix}$	larger	$\begin{pmatrix}1876 & 11 & 963 & 971 & 1970 & 980\end{pmatrix}$
highest	$\begin{pmatrix}1876 & 7 & 797 & 822 & 1950 & 827 & 830 & 841\end{pmatrix}$	largest	$\begin{pmatrix}1876 & 11 & 963 & 971 & 1970 & 953 & 1026\end{pmatrix}$
highly	$\begin{pmatrix}1876 & 7 & 797 & 822 & 1950 & 818 & 831\end{pmatrix}$	largely	$\begin{pmatrix}1876 & 11 & 963 & 971 & 1970 & 979 & 954\end{pmatrix}$

TABLE 6.1: Comparison of Similar English Words and their States

It can be seen that words with the same states also share meaning; however, although the adjectival suffices (for instance *high-est*/*large-st* and *high-ly*/*large-ly*) are the same, their states are not. Therefore, morphological information is not expected to be incorporated by single states. Even if words share the same states, this does not mean they are similar: Besides *highly*, *health*, *heavy*, *hillary*, *healthy*, and *highway* all share state 818. So, despite the valid hypothesis that the number of states becomes manageable through relevant states, the idea of state embeddings has to be discarded. Thus, the analysis in this chapter only focusses on word embeddings. Exceptional results, such as for antonymy and entailment questions, are best explained by coincidence.

Though word embeddings perform better, they do not match the results achieved in the original WORD2VEC, GLOVE and FASTTEXT publications. The largest difference between those papers and this thesis lies in the amount of data, which is up to 30,000, 400,000, and presumably $\approx 30,000$ words per vocabulary, respectively, compared to 2000 words used here. Even worse than the sheer size of those vocabularies is the ratio of functional to lexical words. Since the training set in this study comprises of the most common words, especially in the small batches, containing 500 and 1000 words, this ratio is inconveniently high, because functional words are among the most frequent (see appendices Appendix A and Appendix B). Distributional semantics approaches model meaning through context, which is why results can be easily diluted by terms that bear functional, though no lexical information. Therefore, only results for the largest vocabulary are discussed. Gains for smaller vocabularies are also interpreted as random error.

Furthermore, the most common top five similar words for OOV terms in Figure 5.3 have no resemblance among each other. Together with the distribution of answers to the analogical questions in Figure 5.3, this indicates a hubness problem, where certain words appear disproportionally often. Strangely, these words are in many cases the rarest ones in the vocabularies (cf. Figure 5.3). This might be because the most infrequent words also have the most vague vector representation, due to their least number of training samples.

Context windows are comparable to those in other approaches, and results in Figure 5.6 and Figure 5.7 suggest that larger sizes would worsen the performance. Similarly, embedding dimension sizes do also not account for the downfall; relative ranks in figures 5.8 and 5.9 reveal a clear optimum for small, i.e. five, dimensions, although results are slightly surging for 20. This is not surprising, given that Pennington et al. (2014) originally use 300 - 1000 dimensions for a 200-times larger vocabulary (400,000 words) and a more than 1000-times bigger corpus (42 billion words). That results roughly in a ratio of one dimension per 400 to 1,300 words. Following their empirical investigation, five embedding dimensions would already pose an upper bound for a vocabulary of 2000 words.

This leads to the evaluation of analogical questions (cf. Figures 5.10 and 5.11). Outcomes for gender questions are roughly on par for both languages. German can predict singular, plural, and declinated forms more effectively by determiners (cf. *die Länder* - *den Ländern* and *the country* - *the countries*), which therefore yields a much higher median for declination questions than in English has for singular/ plural. Derivational questions are answered the closest to correct, possibly because the words in the set (for instance, *drive - driver*) are limited the certain contexts. Following this reasoning, questions asking to conjugate are answered worse in both languages, probably due to the fact that the verbs in the test set are not determined to specific contexts (e.g. *take*, *go*, *do*). Although the adjectives in the test set are rather generic (such as *good*, *bad*, *large*, *small*), they are on average more infrequent than verbs, especially in German (cf. tables A.1 and B.1), such that they are *statistically* bounded to fewer contexts, making them easier to distinguish. This might explain the comparably high relative ranks for distinction/ comparison and comparison/ declination, respectively. Distinct, but similar contexts of proper nouns could also be the reason for relatively good ratings for the antonymy/ entailment tasks.

The last plots with regards to word vectors (see box-plots 5.12) refute another hypothesis, namely that larger dictionaries provide more context and thus more disambiguation. The results for analogical questions in the 500 words vocabulary decrease with more words, while results for questions in the mid-size vocabulary remain equal when employing 2000 terms. However, the idea itself is not wrong, as tables 3.6 (WORD2VEC) and 3.8 (GLOVE), as well as 3.9 (FASTTEXT) demonstrate. The expected benefit of GLOVE, namely its explicit use of global corpora statistics, can neither be verified, nor rejected.

Moving on to the translations, the first thing to note is that results are better than the ones for monolingual word vectors. Partially, the increase in relevant ranks could be accounted to the evaluation process: Whereas the analogical questions accept one answer only, the translational evaluation accepts all word forms of the same lemma. But apparently, the erroneous information is incorporated into the embeddings in such a consistent way that it can be exploited by the translational process. This is viewed as a general proof of concept of the robustness of TRANSRANK.

Beginning with the number of iterations until convergence, it is evident (cf. figures 5.14 and 5.15) that densely incorporated information in low-dimensional vector spaces take longer to align than underspecified larger embedding dimensions. The larger the vocabulary becomes, the more iterations are necessary. Values range between nine and 17, which is in line with the results by Dorow et al. (2009) (six iterations) and Rothe and Schütze (2014) (five and 20 runs), and a similar reported damping factor.

Albeit not as severe as for the word vectors, the distribution of the ten most common translations in Figure 5.16 again reveals a hubness problem. This issue is also observed by Artetxe et al. (2016). The most common five English-to-German/ German-to-English translations for OOV terms in plots 5.17 emphasize the lack of captioning similarity in the embedding vectors. As chart 5.18 shows, more words from lower-sized vocabularies are selected as answers. This is important, because it demonstrates that the translational process adds some degree of freedom onto the inherent structure of the word vectors. In case of English-to-German, the variation is a bit lower than it is for German-to-English translations.

In contrast to the findings of the monolingual tasks, the quality of translation slowly grows asymptotically with the context size for German-to-English and only slightly decreases for English-to-German (see plots in Figure 5.20). Similarly, embedding sizes have almost no effect on relative ranks (cf. Figure 5.21). For translations into both directions, nouns show the best quality, followed by verbs, adjectives/ adverbs, and lastly proper nouns. Differences between the first three are relatively marginal, whereas proper nouns are strikingly off. This is especially interesting given the results for analogical questions in box-plots 5.10. A reasonable explanation for this phenomenon cannot be offered.

Generally, German-to-English translations are better than English-to-German ones. Morphological variability in German is most probably responsible for this gap; while the English vocabulary contains more different words, German has more different word forms.

Finally, the hypothesis that a larger vocabulary leads to better translations, can be revoked, at least for translations. Terms from both the smallest, as well as the mid-sized dictionary, are better translated in the 1000 and 2000 word vocabularies (see graphs in Figure 5.25). Focussing only on numbers, the approach presented here comes closest to Dorow et al. (2009), with mean (not *median*) relative ranks reaching from 0.205 to 0.313 (Section 5.2.2). However, the similarity among their closest translations is much more pronounced (cf. Table 5.20).

Other methodologies presented here, neither with, nor without bilingual seed dictionaries, could not be competed with. Especially Conneau et al. (2017) demonstrate, how well unsupervised word-to-word translation can perform.

Apart from its results, the main drawback of TRANSRANK's translation process is that the number of steps is bounded by the bi-quadratic size of the input vocabularies; for instance, if the source and the goal dictionary contain each 1,000 words represented in five embedding dimensions, the number of steps *per* update is 25,000,000. In comparison, non-optimized SIMRANK and COSIMRANK frameworks would only take 25,000. Furthermore, the approach in the current stage is not vectorized, i.e. it is not formulated in matrix notation, which can be handled more efficiently by specialized libraries. Therefore, each update has to be computed through pythonic loops, additionally worsening the runtime.

In conclusion, TRANSRANK's declared goal to build a dictionary, both unsupervised and accurate, on a small data set, cannot be fully accomplished with the current set-up. Its potential in terms of relative ranks, as indicated by Table 5.5 and Table 5.6, seems promising, but there certainly remains room for improvement. Therefore, the last section aims to improve the method of this thesis by collecting effective ideas of related work.

6.1.2 The broader Picture

But before concluding the thesis with ideas for future explorations, it is worth taking a break to localize the project's current position in the landscape of automated translation. The classic procedure starts by acquiring vast amounts of parallel corpora. Based on those, a bilingual language model is constructed, which predicts the translation of the current word based on previous (and sometimes subsequent) terms. This is either done in one step (for instance, using bi-directional recurrent NNs (Koehn, 2017) page 48), or split into two, where a "table that associates a real number between zero and one with every possible pairing" between source and goal language is first compiled and then serves as look-up for a statistical process (cf. (Brown et al., 1990)).

This thesis aimed to remedy the main drawback of both methodologies, namely their reliance on parallel data, which involves tedious and expensive work of human translators. Especially for languages with fewer resources, unsupervised strategies are desirable. With word vectors of arbitrary number and dimension as input, alignments are calculated in-between, resulting in a ready-to-use bilingual lookup-table. Although the proposed method revealed the aforementioned drawbacks, the improvement on defective word vectors demonstrated the robustness of the program. The reason for the system's resilience to erroneous input is thought to be the novel integration of missing context (see Section 4.4). That is why, further research in the area seems promising.

On the path towards a functioning MT framework, the next stepping stone would consist of monolingual language models, which can be combined with the alignments. This way, the barrier between languages could be at least partially overcome in unsupervised manner.

6.2 Future Work

The discussion shows that there is enough room for improvement. This section therefore gives some ideas for future work.

Proper Selection of Words

One disadvantage of the project was the unsupervised, though blind selection of words for the test set, which leads to an over-representation of functional words. A method such as *tf.idf* (3.11) could succeed in filtering non-lexical and uninformative words, for example when applied to the singular sentences in the WORTSCHATZ corpus.

More Training Epochs

Another way to improve the embedding vectors is to extend the number of training epochs. This project followed the recommendations of Pennington et al. (2014) and used 50 iterations, as the number of dimensions in all experiments was below 300. However, due to the high number of too generic words, a closer error margin might be necessary.

Incorporation of Sub-Word Information

The outstanding results of Conneau et al. (2017) emphasize the importance of morphological information. Hence, instead of state embeddings, tried-and-tested FASTTEXT vectors can be employed to boost the performance.

Pre-informed Seed Dictionary

The approach of Artetxe et al. (2017) shows how a seed dictionary can be implemented in unsupervised fashion, by aligning all equal numerical strings. Haghighi et al. (2008) and Mikolov et al. (2013) elaborate on this and use orthographic features to enhance translation quality. All of these methods show better results than the one presented here, thus, initializing the similarity matrix with, for example, edit distance ought to improve the translation quality.

Reformulation in Matrix Notation

The translation process suffered from an exorbitantly high run-time, mainly because it could not be rewritten in matrix notation. Integrating missing context requires a distance metric between the entries, which cannot be captured by matrix calculus. One way to mitigate this could be to calculate similarity matrices two times: First, the word vectors are normalized by the softmax-function, yielding a probability distribution for every row in the embedding matrices. So, the regular SIMRANK computation can be executed using vectorization. Second, as this captures only correspondences between the largest embeddings, the process is repeated, this time with *inverted* embedding values, multiplied by -1.
After each iteration, the two matrices for word alignments and the two matrices containing the embedding alignments could be averaged, respectively.

Application of Procruste's Solution

Haghighi et al. (2008) successfully apply Procruste's solution as refinement to their calculated translation matrix. This might also improve the results .

Enforced Re-Translation

An alternative to Procruste's solution could be the following idea. It formalizes the intuition that a word w_i's translation should re-translate back to w_i:

$$\begin{aligned} \mathbf{S}_w^\star &= arg \min_{\mathbf{S}_e} \sum_{w_i} \left\| \mathbf{x}_i \mathbf{S}_e \mathbf{S}_e^T - \mathbf{x}_i \right\|_2^2 \\ &= arg \min_{\mathbf{S}_e} \left\| \mathbf{X} \mathbf{S}_e \mathbf{S}_e^T - \mathbf{X} \right\|_2^2 \end{aligned} \tag{6.1}$$

where $\mathbf{X}$ is the embedding matrix for the source language, and $\mathbf{S}_e$ the similarity matrix embeddings. Analogously, for the target language with embedding matrix $\mathbf{Y}$:

$$\mathbf{S}_e^\star = arg \min_{\mathbf{S}_e} \left\| \mathbf{Y} \mathbf{S}_e^T \mathbf{S}_e - \mathbf{Y} \right\|_2^2 \tag{6.2}$$

The same procedure can be applied to the word-similarity matrix $\mathbf{S}_w$.
Being similar to the approach of Cisse et al. (2017) (cf. formula (4.39)) in which the translation matrix is regularized towards orthogonality, equations (6.1) and

(6.2) allow more freedom, as the number of both the number of words embeddings do not need to correspond.

Tackling the Hubness-Problem

One prevalent problem throughout the evaluation of word vectors and translation matrices was that certain words - mostly infrequent ones - were overrepresented in the results. As turned out, cosine similarity is robust towards vector length, but not towards hubs. Therefore, it is recommended for future investigations to adept *csls* (see equation (4.39)) by Conneau et al. (2017) to tackle the hubness problem.

Improved Evaluation

The current evaluation uses one-dimensional box-plots to analyze the influences of parameters. For future work, a more detailed investigation could employ, for instance, regression models to examine how parameters affect each other.

Regarding the test set for translations, a more fine-grained translation procedure could involve exact word forms, than only the correct lemmata.

Appendix A

English Vocabulary

This Appendix lists the 2000 most common words in the English lexicon, sorted accordingly to their frequency rank, as well as the data subset used for evaluation.

0 − 50		50 − 100		100 − 150		150 − 200		200 − 250	
0		50	which	100	those	150	including	200	thursday
1	the	51	when	101	make	151	both	201	used
2	to	52	out	102	since	152	another	202	tuesday
3	of	53	would	103	day	153	good	203	each
4	and	54	her	104	even	154	part	204	days
5	in	55	all	105	during	155	help	205	support
6	that	56	what	106	being	156	here	206	united
7	for	57	if	107	city	157	family	207	wednesday
8	is	58	than	108	three	158	found	208	later
9	on	59	year	109	against	159	public	209	done
10	it	60	two	110	president	160	right	210	top
11	said	61	some	111	any	161	need	211	month
12	with	62	so	112	say	162	team	212	every
13	was	63	first	113	still	163	under	213	got
14	he	64	time	114	through	164	man	214	five
15	at	65	no	115	made	165	school	215	china
16	as	66	over	116	home	166	game	216	big
17	from	67	just	117	according	167	life	217	children
18	are	68	other	118	then	168	national	218	number
19	be	69	last	119	down	169	friday	219	took
20	have	70	into	120	these	170	court	220	party
21	by	71	like	121	million	171	four	221	ago
22	has	72	years	122	way	172	second	222	money
23	this	73	says	123	off	173	house	223	however
24	but	74	our	124	news	174	states	224	case
25	his	75	them	125	work	175	report	225	minister
26	an	76	could	126	week	176	end	226	lot
27	they	77	police	127	going	177	called	227	few
28	we	78	do	128	company	178	really	228	south
29	not	79	now	129	take	179	same	229	several
30	will	80	your	130	re	180	business	230	international
31	you	81	get	131	very	181	left	231	deal
32	who	82	state	132	much	182	former	232	put
33	more	83	how	133	me	183	show	233	obama
34	their	84	my	134	see	184	use	234	market
35	were	85	while	135	should	185	times	235	without
36	been	86	only	136	around	186	women	236	months
37	or	87	most	137	country	187	come	237	ve
38	had	88	before	138	did	188	own	238	came
39	about	89	many	139	think	189	best	239	killed
40	one	90	because	140	well	190	monday	240	never
41	she	91	world	141	group	191	place	241	night
42	after	92	him	142	next	192	health	242	york
43	new	93	percent	143	between	193	officials	243	past
44	its	94	where	144	such	194	security	244	change
45	people	95	told	145	know	195	long	245	added
46	also	96	government	146	photos	196	high	246	statement
47	there	97	back	147	don	197	view	247	season
48	up	98	may	148	go	198	set	248	won
49	can	99	us	149	want	199	too	249	little

250 – 300		300 – 350		350 – 400		400 – 450		450 – 500	
250	billion	300	power	350	points	400	died	450	final
251	things	301	hours	351	oil	401	july	451	within
252	better	302	able	352	once	402	announced	452	apple
253	area	303	making	353	students	403	late	453	feel
254	sunday	304	working	354	attack	404	job	454	full
255	why	305	earlier	355	syria	405	games	455	policy
256	until	306	data	356	program	406	looking	456	control
257	early	307	didn	357	press	407	started	457	hard
258	office	308	six	358	june	408	person	458	old
259	away	309	went	359	head	409	decision	459	given
260	already	310	start	360	live	410	care	460	doesn
261	service	311	per	361	give	411	taking	461	growth
262	law	312	expected	362	small	412	officers	462	clear
263	might	313	men	363	enough	413	meeting	463	forces
264	american	314	county	364	john	414	half	464	russia
265	look	315	federal	365	must	415	official	465	reports
266	information	316	run	366	plan	416	march	466	event
267	today	317	though	367	services	417	rights	467	series
268	something	318	companies	368	call	418	issue	468	plans
269	least	319	white	369	important	419	close	469	online
270	media	320	asked	370	social	420	often	470	together
271	war	321	recent	371	future	421	director	471	east
272	story	322	center	372	free	422	global	472	process
273	local	323	taken	373	known	423	doing	473	financial
274	military	324	keep	374	trying	424	face	474	europe
275	among	325	washington	375	released	425	road	475	board
276	department	326	again	376	canada	426	history	476	america
277	saturday	327	young	377	shot	427	real	477	iran
278	video	328	win	378	behind	428	economic	478	force
279	death	329	less	379	having	429	leader	479	ever
280	great	330	point	380	getting	430	coming	480	gallery
281	play	331	street	381	move	431	weeks	481	human
282	does	332	major	382	chief	432	nearly	482	watch
283	members	333	yet	383	clinton	433	bank	483	september
284	fire	334	countries	384	held	434	european	484	began
285	north	335	hit	385	trump	435	hospital	485	agency
286	across	336	car	386	attacks	436	officer	486	park
287	whether	337	black	387	using	437	believe	487	led
288	reported	338	always	388	morning	438	germany	488	almost
289	water	339	saying	389	food	439	side	489	due
290	system	340	third	390	authorities	440	general	490	further
291	far	341	different	391	along	441	comes	491	research
292	seen	342	photo	392	woman	442	continue	492	capital
293	open	343	likely	393	am	443	bill	493	industry
294	find	344	air	394	following	444	economy	494	leaders
295	political	345	others	395	lost	445	thing	495	sales
296	community	346	become	396	west	446	building	496	investigation
297	near	347	pay	397	outside	447	foreign	497	love
298	campaign	348	share	398	lead	448	line	498	record
299	university	349	ll	399	cnn	449	possible	499	sure

500 – 550		550 – 600		600 – 650		650 – 700		700 – 750	
500	large	550	lives	600	special	650	nation	700	customers
501	islamic	551	read	601	spokesman	651	agreement	701	couple
502	january	552	kind	602	isis	652	became	702	currently
503	name	553	available	603	wrote	653	families	703	bush
504	ap	554	needs	604	council	654	low	704	wants
505	shows	555	fact	605	involved	655	potential	705	act
506	april	556	staff	606	someone	656	leave	706	residents
507	film	557	cost	607	related	657	interview	707	fall
508	problem	558	current	608	forward	658	defense	708	result
509	town	559	region	609	union	659	father	709	posted
510	issues	560	running	610	august	660	personal	710	red
511	prime	561	site	611	california	661	nothing	711	email
512	despite	562	iraq	612	france	662	committee	712	changes
513	project	563	facebook	613	friends	663	soon	713	kids
514	thought	564	recently	614	played	664	list	714	london
515	wanted	565	french	615	access	665	questions	715	google
516	prices	566	instead	616	parents	666	middle	716	border
517	body	567	increase	617	phone	667	san	717	scene
518	workers	568	seven	618	refugees	668	happened	718	bad
519	music	569	saw	619	players	669	greece	719	prison
520	comments	570	shooting	620	district	670	trade	720	showed
521	energy	571	means	621	december	671	private	721	interest
522	medical	572	presidential	622	needed	672	career	722	isn
523	minutes	573	visit	623	safety	673	talks	723	worked
524	study	574	release	624	race	674	legal	724	british
525	let	575	provide	625	goal	675	college	725	cup
526	makes	576	return	626	areas	676	gave	726	stay
527	stop	577	try	627	quarter	677	average	727	total
528	strong	578	role	628	st	678	summer	728	whose
529	tax	579	mother	629	everyone	679	league	729	chance
530	rate	580	include	630	star	680	meet	730	justice
531	paris	581	charges	631	higher	681	jobs	731	similar
532	space	582	content	632	tv	682	matter	732	drug
533	republican	583	received	633	arrested	683	violence	733	rather
534	front	584	post	634	anything	684	age	734	syrian
535	groups	585	hope	635	website	685	education	735	himself
536	price	586	latest	636	cut	686	remains	736	church
537	election	587	level	637	ahead	687	thousands	737	seattle
538	mr	588	key	638	russian	688	idea	738	previous
539	child	589	action	639	everything	689	question	739	charged
540	order	590	member	640	eight	690	son	740	problems
541	course	591	comment	641	david	691	room	741	judge
542	vote	592	november	642	biggest	692	senior	742	difficult
543	based	593	fight	643	cases	693	situation	743	japan
544	inside	594	chinese	644	dead	694	anyone	744	coach
545	experience	595	october	645	crisis	695	twitter	745	growing
546	conference	596	bring	646	although	696	ground	746	debate
547	technology	597	german	647	living	697	especially	747	image
548	actually	598	risk	648	sent	698	main	748	turn
549	central	599	field	649	development	699	executive	749	wife

750 − 800		800 − 850		850 − 900		900 − 950		950 − 1000	
750	stories	800	army	850	pressure	900	reporters	950	senate
751	playing	801	climate	851	daily	901	james	951	businesses
752	americans	802	heard	852	cause	902	driver	952	includes
753	miles	803	cbc	853	western	903	test	953	toward
754	incident	804	themselves	854	closed	904	investors	954	form
755	rates	805	costs	855	student	905	cars	955	sometimes
756	period	806	mark	856	michael	906	player	956	crime
757	offer	807	trial	857	manager	907	safe	957	peace
758	lower	808	club	858	province	908	threat	958	appeared
759	island	809	example	859	popular	909	stage	959	seems
760	weekend	810	nine	860	candidate	910	budget	960	success
761	station	811	whole	861	quickly	911	crash	961	performance
762	leading	812	canadian	862	light	912	firm	962	raised
763	hold	813	football	863	sign	913	opposition	963	cannot
764	create	814	organization	864	adding	914	ceo	964	opened
765	eu	815	reach	865	northern	915	ministry	965	parts
766	evidence	816	movie	866	above	916	app	966	worth
767	book	817	travel	867	serious	917	land	967	effort
768	helped	818	turkey	868	build	918	included	968	moment
769	associated	819	nuclear	869	speech	919	sea	969	starting
770	training	820	longer	870	wall	920	review	970	reached
771	talk	821	calls	871	administration	921	los	971	attention
772	efforts	822	search	872	spending	922	ready	972	moved
773	attorney	823	takes	873	allowed	923	rise	973	happen
774	turned	824	credit	874	ukraine	924	numbers	974	message
775	fans	825	internet	875	opportunity	925	decided	975	brown
776	reuters	826	brought	876	pretty	926	concerns	976	israel
777	victims	827	immediately	877	employees	927	cancer	977	king
778	wasn	828	either	878	candidates	928	felt	978	itself
779	bit	829	fighting	879	works	929	rules	979	de
780	results	830	weather	880	ways	930	fourth	980	giving
781	stock	831	texas	881	details	931	majority	981	updated
782	short	832	met	882	mobile	932	southern	982	rose
783	focus	833	civil	883	follow	933	patients	983	india
784	february	834	network	884	paid	934	green	984	goes
785	congress	835	heart	885	africa	935	dr	985	terms
786	remain	836	users	886	secretary	936	images	986	mayor
787	tell	837	store	887	drive	937	investment	987	address
788	allow	838	annual	888	markets	938	rest	988	angeles
789	gas	839	impact	889	loss	939	com	989	sports
790	largest	840	vehicle	890	step	940	injured	990	society
791	probably	841	published	891	contact	941	migrants	991	commission
792	response	842	products	892	homes	942	source	992	missing
793	tried	843	position	893	file	943	break	993	cent
794	fell	844	reason	894	victory	944	created	994	sense
795	nations	845	single	895	conditions	945	match	995	simply
796	democratic	846	accused	896	hundreds	946	round	996	korea
797	spent	847	buy	897	page	947	emergency	997	common
798	hand	848	schools	898	huge	948	named	998	traffic
799	paul	849	events	899	mean	949	association	999	ended

1000 − 1050		1050 − 1100		1100 − 1150		1150 − 1200		1200 − 1250	
1000	gun	1050	tour	1100	base	1150	mind	1200	toronto
1001	expect	1051	australia	1101	compared	1151	served	1201	effect
1002	below	1052	check	1102	property	1152	goals	1202	device
1003	failed	1053	knew	1103	particularly	1153	learn	1203	eventually
1004	opening	1054	true	1104	republicans	1154	hear	1204	version
1005	england	1055	else	1105	disease	1155	greek	1205	citizens
1006	looks	1056	certain	1106	track	1156	resources	1206	positive
1007	flight	1057	forced	1107	protect	1157	windows	1207	records
1008	charge	1058	suspect	1108	model	1158	hopes	1208	approach
1009	ball	1059	followed	1109	revenue	1159	leadership	1209	ruling
1010	treatment	1060	target	1110	provided	1160	range	1210	damage
1011	continued	1061	hands	1111	via	1161	spoke	1211	contributed
1012	demand	1062	population	1112	vehicles	1162	husband	1212	returned
1013	voters	1063	train	1113	leaving	1163	politics	1213	parties
1014	murder	1064	cities	1114	preferences	1164	additional	1214	figure
1015	uk	1065	decades	1115	hour	1165	believed	1215	quality
1016	florida	1066	friend	1116	injuries	1166	begin	1216	camp
1017	account	1067	words	1117	parliament	1167	mission	1217	consider
1018	daughter	1068	moving	1118	beat	1168	hall	1218	bay
1019	significant	1069	feet	1119	issued	1169	declined	1219	weapons
1020	happy	1070	production	1120	maybe	1170	arrived	1220	born
1021	understand	1071	scored	1121	runs	1171	investigators	1221	victim
1022	challenge	1072	caused	1122	product	1172	hearing	1222	relationship
1023	gets	1073	eastern	1123	shares	1173	television	1223	girls
1024	agreed	1074	wrong	1124	cash	1174	picture	1224	heavy
1025	debt	1075	raise	1125	alone	1175	lack	1225	exchange
1026	criminal	1076	beyond	1126	considered	1176	grand	1226	culture
1027	fund	1077	ask	1127	meanwhile	1177	chris	1227	storm
1028	killing	1078	class	1128	chicago	1178	talking	1228	avoid
1029	launched	1079	gone	1129	science	1179	host	1229	blood
1030	conservative	1080	afternoon	1130	calling	1180	practice	1230	communities
1031	amount	1081	pass	1131	saudi	1181	measures	1231	kept
1032	driving	1082	pm	1132	changed	1182	scheduled	1232	systems
1033	stand	1083	girl	1133	troops	1183	environment	1233	speaking
1034	claims	1084	centre	1134	trip	1184	poor	1234	african
1035	easy	1085	river	1135	coalition	1185	boy	1235	militants
1036	management	1086	levels	1136	funding	1186	signed	1236	devices
1037	plane	1087	addition	1137	income	1187	offered	1237	ensure
1038	quite	1088	previously	1138	evening	1188	looked	1238	guard
1039	airport	1089	williams	1139	passed	1189	ability	1239	banks
1040	coast	1090	described	1140	article	1190	noted	1240	strategy
1041	operations	1091	hotel	1141	entire	1191	continues	1241	construction
1042	planned	1092	throughout	1142	date	1192	democrats	1242	cover
1043	confirmed	1093	researchers	1143	marriage	1193	complete	1243	mexico
1044	increased	1094	save	1144	governor	1194	battle	1244	helping
1045	experts	1095	conflict	1145	afp	1195	smith	1245	perhaps
1046	teams	1096	overall	1146	winning	1196	brother	1246	soldiers
1047	built	1097	britain	1147	seeking	1197	dropped	1247	natural
1048	art	1098	finally	1148	condition	1198	launch	1248	prosecutors
1049	value	1099	sold	1149	blue	1199	george	1249	snow

1250 − 1300		1300 − 1350		1350 − 1400		1400 − 1450		1550 − 1500	
1250	programs	1300	finished	1350	drivers	1400	foundation	1450	lee
1251	term	1301	winter	1351	owner	1401	thinking	1451	wouldn
1252	title	1302	armed	1352	opinion	1402	restaurant	1452	spring
1253	sexual	1303	lawyer	1353	straight	1403	suffered	1453	lines
1254	arrest	1304	appears	1354	choice	1404	successful	1454	massive
1255	design	1305	carolina	1355	join	1405	beach	1455	flag
1256	software	1306	send	1356	progress	1406	concern	1456	gov
1257	tough	1307	holding	1357	word	1407	appeal	1457	voice
1258	fear	1308	places	1358	supreme	1408	decade	1458	sector
1259	barack	1309	guilty	1359	planning	1409	extra	1459	primary
1260	improve	1310	insurance	1360	streets	1410	answer	1460	paper
1261	blog	1311	sell	1361	female	1411	click	1461	nearby
1262	elections	1312	finance	1362	remember	1412	broke	1462	reasons
1263	regional	1313	caught	1363	seeing	1413	putin	1463	dangerous
1264	johnson	1314	signs	1364	add	1414	offers	1464	japanese
1265	appear	1315	muslim	1365	supporters	1415	oct	1465	legislation
1266	certainly	1316	joined	1366	showing	1416	powerful	1466	jail
1267	competition	1317	contract	1367	dog	1417	illegal	1467	afghanistan
1268	lake	1318	dollars	1368	push	1418	francisco	1468	putting
1269	radio	1319	bbc	1369	highest	1419	grow	1469	deep
1270	super	1320	drop	1370	un	1420	request	1470	relations
1271	door	1321	features	1371	enforcement	1421	protection	1471	sources
1272	drugs	1322	original	1372	traditional	1422	doctors	1472	figures
1273	receive	1323	fed	1373	hillary	1423	funds	1473	museum
1274	housing	1324	vice	1374	ran	1424	carried	1474	various
1275	beginning	1325	watching	1375	faces	1425	aircraft	1475	approved
1276	walk	1326	intelligence	1376	wait	1426	projects	1476	al
1277	benefits	1327	believes	1377	hill	1427	struck	1477	block
1278	movement	1328	martin	1378	sex	1428	refugee	1478	regular
1279	ice	1329	gold	1379	chairman	1429	rule	1479	affected
1280	microsoft	1330	asking	1380	seem	1430	limited	1480	benefit
1281	independent	1331	yards	1381	operation	1431	christmas	1481	sanders
1282	note	1332	festival	1382	fighters	1432	domestic	1482	trust
1283	mostly	1333	assault	1383	greater	1433	square	1483	reduce
1284	laws	1334	fired	1384	crowd	1434	bus	1484	boston
1285	ones	1335	critical	1385	claim	1435	earth	1485	survey
1286	scott	1336	aid	1386	deputy	1436	fun	1486	poll
1287	ban	1337	designed	1387	stocks	1437	please	1487	unit
1288	identified	1338	worst	1388	mike	1438	miss	1488	individual
1289	multiple	1339	spot	1389	facing	1439	rain	1489	labor
1290	pope	1340	digital	1390	gop	1440	donald	1490	rebels
1291	responsible	1341	millions	1391	couldn	1441	fast	1491	analysts
1292	freedom	1342	jan	1392	religious	1442	warned	1492	truck
1293	required	1343	fair	1393	serve	1443	concerned	1493	operating
1294	baby	1344	consumers	1394	estimated	1444	award	1494	necessary
1295	reality	1345	present	1395	attempt	1445	jones	1495	simple
1296	letter	1346	mass	1396	terrorist	1446	sen	1496	activity
1297	speak	1347	filed	1397	injury	1447	tech	1497	robert
1298	spend	1348	stopped	1398	aren	1448	exactly	1498	proposed
1299	claimed	1349	prevent	1399	yes	1449	consumer	1499	deaths

1500 – 1550		1550 – 1600		1600 – 1650		1650 – 1700		1700 – 1750	
1500	becoming	1550	rising	1600	institute	1650	eyes	1700	advantage
1501	newspaper	1551	authority	1601	negotiations	1651	normal	1701	location
1502	policies	1552	beijing	1602	twice	1652	books	1702	analyst
1503	guy	1553	yemen	1603	holiday	1653	measure	1703	valley
1504	announcement	1554	selling	1604	stars	1654	village	1704	studies
1505	responsibility	1555	asia	1605	animals	1655	spread	1705	ship
1506	waiting	1556	century	1606	bridge	1656	haven	1706	championship
1507	box	1557	guys	1607	sites	1657	starts	1707	complex
1508	dollar	1558	offering	1608	shared	1658	suggested	1708	bowl
1509	completely	1559	card	1609	thomas	1659	knows	1709	largely
1510	alleged	1560	focused	1610	towards	1660	standing	1710	effective
1511	rescue	1561	dec	1611	particular	1661	missed	1711	bomb
1512	zone	1562	index	1612	denied	1662	highway	1712	journal
1513	options	1563	fox	1613	written	1663	feature	1713	elected
1514	strike	1564	sun	1614	brand	1664	governments	1714	accident
1515	sanctions	1565	passengers	1615	pictures	1665	double	1715	cross
1516	usually	1566	professor	1616	cold	1666	ryan	1716	clearly
1517	stores	1567	actions	1617	god	1667	hurt	1717	liberal
1518	gives	1568	code	1618	nov	1668	audience	1718	providing
1519	camera	1569	tournament	1619	amazon	1669	euro	1719	youth
1520	band	1570	meant	1620	earnings	1670	floor	1720	ordered
1521	scientists	1571	losing	1621	lose	1671	platform	1721	shown
1522	thanks	1572	size	1622	nature	1672	names	1722	seconds
1523	francis	1573	iphone	1623	challenges	1673	remained	1723	whom
1524	nbc	1574	suicide	1624	apps	1674	euros	1724	carrying
1525	learned	1575	gay	1625	jury	1675	allegations	1725	solution
1526	amid	1576	feeling	1626	separate	1676	division	1726	specific
1527	equipment	1577	committed	1627	originally	1677	fine	1727	net
1528	sound	1578	fellow	1628	strikes	1678	paying	1728	videos
1529	hot	1579	healthy	1629	turkish	1679	onto	1729	walking
1530	trading	1580	piece	1630	respond	1680	individuals	1730	reform
1531	directly	1581	kill	1631	spokeswoman	1681	nfl	1731	van
1532	pick	1582	keeping	1632	items	1682	seemed	1732	larger
1533	bid	1583	steps	1633	iowa	1683	bodies	1733	screen
1534	sides	1584	analysis	1634	activities	1684	supply	1734	pulled
1535	israeli	1585	terror	1635	discovered	1685	happens	1735	perfect
1536	fully	1586	difference	1636	standard	1686	android	1736	favorite
1537	immigration	1587	discuss	1637	labour	1687	partners	1737	expressed
1538	documents	1588	cuts	1638	located	1688	active	1738	crew
1539	views	1589	author	1639	warning	1689	english	1739	joe
1540	carry	1590	terrorism	1640	journalists	1690	plus	1740	speed
1541	partner	1591	finding	1641	commercial	1691	taxes	1741	avenue
1542	marketing	1592	computer	1642	sept	1692	modern	1742	cameron
1543	owners	1593	wearing	1643	giant	1693	marijuana	1743	revealed
1544	lawmakers	1594	inc	1644	fifth	1694	plays	1744	davis
1545	reporter	1595	older	1645	professional	1695	facility	1745	allows
1546	magazine	1596	yesterday	1646	winner	1696	sheriff	1746	brain
1547	rock	1597	mental	1647	bar	1697	sport	1747	eye
1548	fifa	1598	type	1648	status	1698	tom	1748	steve
1549	protests	1599	sale	1649	shots	1699	awards	1749	richard

1750 – 1800		1800 – 1850		1850 – 1900		1900 – 1950		1950 – 2000	
1750	direct	1800	penalty	1850	port	1900	rubio	1950	testing
1751	abuse	1801	secret	1851	fresh	1901	song	1951	christian
1752	upon	1802	arabia	1852	buildings	1902	cuba	1952	offensive
1753	language	1803	bringing	1853	route	1903	posts	1953	secure
1754	jordan	1804	carson	1854	influence	1904	falling	1954	transportation
1755	polls	1805	joint	1855	pakistan	1905	premier	1955	closer
1756	beautiful	1806	treated	1856	reading	1906	uses	1956	egypt
1757	myself	1807	require	1857	infrastructure	1907	writer	1957	oregon
1758	agencies	1808	philadelphia	1858	path	1908	aware	1958	doctor
1759	moscow	1809	telling	1859	organizations	1909	politicians	1959	hong
1760	peter	1810	hoping	1860	increasing	1910	investigating	1960	fish
1761	lived	1811	crimes	1861	protesters	1911	generation	1961	existing
1762	champion	1812	walker	1862	unique	1912	targets	1962	expensive
1763	sister	1813	pair	1863	dark	1913	slow	1963	positions
1764	lawsuit	1814	decisions	1864	highly	1914	feb	1964	fbi
1765	rally	1815	display	1865	fuel	1915	affairs	1965	initially
1766	faced	1816	physical	1866	determine	1916	easier	1966	buying
1767	iraqi	1817	minute	1867	heat	1917	quick	1967	session
1768	territory	1818	stadium	1868	jersey	1918	surprise	1968	ideas
1769	option	1819	yourself	1869	declared	1919	taiwan	1969	et
1770	kim	1820	ongoing	1870	grew	1920	merkel	1970	senator
1771	aug	1821	smaller	1871	corporate	1921	serving	1971	targeted
1772	rare	1822	ben	1872	animal	1922	enjoy	1972	voted
1773	dozens	1823	suspected	1873	reportedly	1923	cbs	1973	smart
1774	explained	1824	allegedly	1874	ultimately	1924	edition	1974	teachers
1775	muslims	1825	sort	1875	suspects	1925	prepared	1975	royal
1776	willing	1826	bigger	1876	moments	1926	accept	1976	draw
1777	coverage	1827	opportunities	1877	completed	1927	weight	1977	patient
1778	boat	1828	visitors	1878	basis	1928	occurred	1978	clients
1779	finish	1829	placed	1879	agree	1929	korean	1979	loved
1780	responded	1830	standards	1880	everybody	1930	initial	1980	arms
1781	score	1831	attend	1881	shortly	1931	boys	1981	linked
1782	summit	1832	customer	1882	sentence	1932	apartment	1982	forecast
1783	convicted	1833	boost	1883	viewed	1933	bought	1983	pilot
1784	seek	1834	plant	1884	table	1934	nice	1984	removed
1785	gmt	1835	sitting	1885	wounded	1935	italy	1985	gain
1786	developed	1836	clean	1886	votes	1936	corruption	1986	raising
1787	subject	1837	prior	1887	bottom	1937	apparently	1987	wilson
1788	decide	1838	activists	1888	bond	1938	wind	1988	expand
1789	historic	1839	proposal	1889	thank	1939	shop	1989	collection
1790	cruz	1840	environmental	1890	character	1940	tells	1990	core
1791	hasn	1841	develop	1891	tests	1941	enter	1991	journey
1792	provides	1842	managed	1892	confidence	1942	criticism	1992	wars
1793	respect	1843	whatever	1893	seat	1943	hate	1993	choose
1794	accounts	1844	none	1894	extremely	1944	learning	1994	changing
1795	creating	1845	wide	1895	indian	1945	broken	1995	developing
1796	pain	1846	actor	1896	downtown	1946	tweet	1996	sentenced
1797	australian	1847	worse	1897	critics	1947	speaks	1997	welcome
1798	increasingly	1848	colorado	1898	web	1948	berlin	1998	carter
1799	allowing	1849	protest	1899	connection	1949	refused	1999	hits

PoS	Test Case	Lemma	Frequency Rank ≤500	Frequency Rank ≤1000	Frequency Rank ≤2000
Noun	Gender Singular/ Plural	male	man men		boy boys
		female	woman women		girl girls
	Singular/ Plural	day	day days		
		year	year years		
		country	country countries		
	Derivation	player		player players	
		driver		driver	drivers
		development		development	
		movement			movement
Proper Noun	Entailment	America	America		
		Washington	Washington		
		Russia	Russia		
		Moscow			Moscow
		Europe	Europe		
		England			England
		London		London	
		France		France	
		Paris		Paris	
Verb	Conjugation	(to) take	take taking took taken	takes	
		(to) go	go going went	goes	gone
		(to) have	have having has had		haven hasn
		(to) do	do does doesn don doing did didn done		
		(to) play	play	played playing	plays
		(to) show	show	showed shows	shown showing
		(to) make	make making made	makes	
	Derivation	(to) drive		drive	
		(to) move	move	moved	
		(to) think	think	thought	thinking
		(to) feel	feel	felt	feeling
		(to) help	help	helped	helping
		(to) develop			develop developed developing
		(to) increase		increase	increasing increased
		(to) report	report reported		
Adjectives Adverbs	Distinction Comparative Derivation Antonymy	large		large largest	larger largely
		high	high	higher	highly highest
		small	small		smaller
		low		low lower	
		good	good better best well		
		bad		bad	worse worst
		American	American		
		Russian		Russian	
		European	European		
		reported	reported		reportedly
		increasing			increasing increasingly
Noun/ Adjective	OOV	national	national		
		emergency		emergency	
		infrastructure			infrastructure

TABLE A.1: List of English Words for the Evaluation

Appendix B

German Vocabulary

This Appendix lists lists the data subset, as well as the 2000 most common words from the German lexicon, sorted according to their frequency rank.

0 – 50		50 – 100		100 – 150		150 – 200		200 – 250	
0		50	mehr	100	ihr	150	wer	200	bild
1	die	51	war	101	dabei	151	kein	201	während
2	der	52	man	102	menschen	152	allem	202	ihm
3	und	53	oder	103	ab	153	ganz	203	einfach
4	in	54	sein	104	sehr	154	alles	204	erste
5	das	55	bis	105	deutschland	155	könnte	205	gab
6	den	56	gegen	106	eines	156	dort	206	geld
7	von	57	wenn	107	viele	157	dafür	207	letzten
8	mit	58	kann	108	geht	158	laut	208	fast
9	zu	59	zur	109	drei	159	ihren	209	davon
10	ist	60	wurde	110	mal	160	andere	210	land
11	auf	61	was	111	gut	161	kommt	211	zurück
12	im	62	hatte	112	waren	162	steht	212	fünf
13	für	63	prozent	113	ersten	163	wollen	213	geben
14	ein	64	schon	114	rund	164	wegen	214	artikel
15	es	65	diese	115	uns	165	ihrer	215	ihnen
16	sich	66	dann	116	wurden	166	sondern	216	wohl
17	nicht	67	durch	117	sagt	167	sowie	217	stehen
18	eine	68	können	118	viel	168	dies	218	milliarden
19	auch	69	unter	119	millionen	169	seinem	219	konnte
20	dem	70	euro	120	seiner	170	seien	220	derzeit
21	sie	71	sei	121	neuen	171	lassen	221	sehen
22	des	72	wieder	122	denn	172	also	222	spiel
23	als	73	doch	123	weiter	173	sollte	223	sogar
24	bei	74	soll	124	müssen	174	dieses	224	weg
25	an	75	ihre	125	heute	175	wäre	225	berlin
26	am	76	habe	126	diesem	176	kommen	226	de
27	nach	77	gibt	127	weil	177	welt	227	gehen
28	dass	78	immer	128	worden	178	mich	228	foto
29	er	79	zwei	129	ohne	179	deutsche	229	deutlich
30	hat	80	keine	130	selbst	180	hatten	230	weniger
31	aus	81	vom	131	zeit	181	macht	231	europa
32	wie	82	seit	132	allerdings	182	usa	232	mann
33	werden	83	beim	133	zwischen	183	würde	233	lange
34	um	84	nun	134	ende	184	einmal	234	bisher
35	aber	85	sagte	135	jahre	185	vier	235	einige
36	sind	86	seine	136	etwa	186	zudem	236	gerade
37	wird	87	damit	137	ob	187	nichts	237	regierung
38	noch	88	jahr	138	weitere	188	leben	238	liegt
39	vor	89	alle	139	etwas	189	diesen	239	sieht
40	einen	90	da	140	anderen	190	jedoch	240	platz
41	so	91	bereits	141	seinen	191	polizei	241	woche
42	einem	92	will	142	erst	192	ihn	242	würden
43	einer	93	jetzt	143	ins	193	mir	243	ihrem
44	ich	94	dieser	144	sollen	194	vergangenen	244	stadt
45	haben	95	muss	145	dazu	195	werde	245	kam
46	über	96	jahren	146	ja	196	wo	246	lässt
47	zum	97	neue	147	machen	197	beiden	247	fall
48	nur	98	hier	148	unternehmen	198	zwar	248	deshalb
49	wir	99	uhr	149	deutschen	199	flüchtlinge	249	zeigt

250 − 300		300 − 350		350 − 400		400 − 450		450 − 500	
250	gar	300	schweizer	350	seite	400	november	450	stellt
251	freitag	301	mittwoch	351	hinter	401	du	451	rennen
252	tag	302	russland	352	wollte	402	aller	452	präsident
253	finden	303	eigentlich	353	je	403	zeigen	453	dessen
254	großen	304	statt	354	beispiel	404	offenbar	454	dürfen
255	darauf	305	angaben	355	bayern	405	punkte	455	österreich
256	große	306	fc	356	heißt	406	weit	456	richtig
257	frau	307	welche	357	sagen	407	meine	457	allein
258	bin	308	zuletzt	358	dürfte	408	januar	458	hin
259	thema	309	gegenüber	359	ging	409	dagegen	459	tagen
260	wenig	310	warum	360	recht	410	erwartet	460	direkt
261	insgesamt	311	genau	361	eher	411	informationen	461	internet
262	bleibt	312	donnerstag	362	samstag	412	grund	462	probleme
263	besser	313	team	363	stunden	413	erklärte	463	facebook
264	hätte	314	bleiben	364	damals	414	juni	464	franken
265	hätten	315	dienstag	365	anfang	415	oktober	465	bietet
266	teil	316	vielen	366	eu	416	bank	466	gegeben
267	gewesen	317	gilt	367	erklärt	417	weiterhin	467	inzwischen
268	frage	318	sechs	368	bislang	418	bringen	468	zumindest
269	neben	319	oft	369	erhalten	419	besten	469	bitte
270	ebenfalls	320	darüber	370	gleich	420	jahres	470	kommentare
271	unsere	321	kunden	371	zukunft	421	gehört	471	erneut
272	eigenen	322	schweiz	372	arbeit	422	nehmen	472	treffen
273	natürlich	323	anderem	373	meisten	423	monaten	473	neues
274	möglich	324	wirklich	374	stark	424	findet	474	mindestens
275	zehn	325	könnten	375	dpa	425	tage	475	weiß
276	zweiten	326	wissen	376	daher	426	apple	476	verletzt
277	kaum	327	saison	377	beide	427	märz	477	sieben
278	denen	328	kurz	378	darf	428	zufolge	478	berichtet
279	besonders	329	zunächst	379	fragen	429	paar	479	konnten
280	jeder	330	sicher	380	deren	430	fest	480	möchte
281	bekannt	331	zusammen	381	daten	431	mai	481	aktuelle
282	später	332	außerdem	382	september	432	entwicklung	482	geschichte
283	dollar	333	per	383	bekommen	433	kommenden	483	tatsächlich
284	nie	334	neu	384	kosten	434	online	484	zahl
285	gemacht	335	spielen	385	kommentar	435	nachdem	485	april
286	pro	336	gute	386	blick	436	schwer	486	nutzen
287	allen	337	tun	387	weiteren	437	arbeiten	487	wirtschaft
288	minuten	338	vielleicht	388	auto	438	china	488	leute
289	wochen	339	zuvor	389	problem	439	daran	489	danach
290	sonntag	340	musste	390	the	440	video	490	mitarbeiter
291	kinder	341	ziel	391	einsatz	441	münchen	491	dennoch
292	griechenland	342	frauen	392	halten	442	paris	492	acht
293	schreiben	343	knapp	393	stand	443	of	493	dezember
294	montag	344	nächsten	394	anders	444	folgen	494	google
295	klar	345	mehrere	395	trainer	445	könne	495	juli
296	antwort	346	zahlen	396	windows	446	markt	496	unserer
297	sollten	347	spieler	397	politik	447	hält	497	ganze
298	keinen	348	stellen	398	merkel	448	is	498	überhaupt
299	schnell	349	trotz	399	solche	449	europäischen	499	staat

500 – 550		550 – 600		600 – 650		650 – 700		700 – 750	
500	wichtig	550	vergleich	600	gruppe	650	straße	700	angesichts
501	sieg	551	einigen	601	ebenso	651	spiele	701	experten
502	nachrichten	552	microsoft	602	nimmt	652	gewinnen	702	abend
503	erstmals	553	gestern	603	spielt	653	plus	703	hohen
504	nutzer	554	führt	604	dritten	654	gebe	704	jeweils
505	läuft	555	nacht	605	setzen	655	bereit	705	flüchtlingen
506	rolle	556	aktuellen	606	obwohl	656	oben	706	staaten
507	liegen	557	sorgen	607	jede	657	autos	707	berliner
508	art	558	junge	608	wien	658	ukraine	708	beste
509	jeden	559	kleinen	609	demnach	659	gefunden	709	angst
510	richtung	560	weltweit	610	türkei	660	kamen	710	version
511	trotzdem	561	unser	611	veröffentlicht	661	sowohl	711	neun
512	frankreich	562	künftig	612	melden	662	schreibt	712	runde
513	new	563	scheint	613	michael	663	darunter	713	details
514	geworden	564	sicherheit	614	sonst	664	at	714	versucht
515	medien	565	mittlerweile	615	innerhalb	665	hinaus	715	fällt
516	dank	566	bereich	616	alten	666	groß	716	landes
517	februar	567	sommer	617	kritik	667	unterstützt	717	bieten
518	syrien	568	verfügung	618	erreicht	668	darum	718	wenige
519	familie	569	bringt	619	monate	669	niemand	719	martin
520	wert	570	fans	620	haus	670	hamburg	720	mutter
521	zeitung	571	kleine	621	hingegen	671	möglichkeit	721	anzeige
522	ort	572	teilte	622	internationalen	672	leider	722	alter
523	personen	573	ergebnis	623	spd	673	minute	723	programm
524	neuer	574	bald	624	endlich	674	dadurch	724	sicht
525	männer	575	mannschaft	625	fehlt	675	orf	725	schrieb
526	antworten	576	aktuell	626	spricht	676	fahren	726	tor
527	müsse	577	seines	627	lesen	677	früher	727	kopf
528	bilder	578	namen	628	schritt	678	angebot	728	entwickelt
529	hoch	579	gemeinsam	629	letzte	679	machte	729	tsipras
530	eben	580	beispielsweise	630	start	680	offen	730	jedem
531	preis	581	region	631	chance	681	erreichen	731	bericht
532	sofort	582	studie	632	wochenende	682	fehler	732	stelle
533	eigene	583	kampf	633	handelt	683	völlig	733	hersteller
534	lang	584	länder	634	gerne	684	banken	734	grenze
535	entscheidung	585	ländern	635	gleichzeitig	685	quartal	735	lag
536	leicht	586	titel	636	opfer	686	peter	736	anzeigen
537	setzt	587	folge	637	lösung	687	nächste	737	league
538	frankfurt	588	partei	638	unterstützung	688	twitter	738	gesehen
539	thomas	589	situation	639	rahmen	689	wahl	739	wasser
540	erfolg	590	höhe	640	schaffen	690	jungen	740	frei
541	mein	591	braucht	641	gesagt	691	beginn	741	meinung
542	gebracht	592	verloren	642	genug	692	projekt	742	größten
543	ag	593	führen	643	hand	693	teilen	743	warten
544	gekommen	594	druck	644	gesellschaft	694	vertrag	744	nämlich
545	mio	595	aufgrund	645	eltern	695	jedes	745	punkten
546	august	596	alte	646	aktien	696	bevor	746	system
547	lage	597	hilfe	647	helfen	697	behörden	747	re
548	schließlich	598	mitte	648	meter	698	europäische	748	blieb
549	zweite	599	guten	649	lediglich	699	geplant	749	cdu

750 – 800		800 – 850		850 – 900		900 – 950		950 – 1000	
750	gericht	800	länger	850	meinen	900	kraft	950	teams
751	sport	801	einzige	851	unseren	901	versuchen	951	interesse
752	bevölkerung	802	zahlreiche	852	bestätigt	902	wichtige	952	moment
753	bürger	803	interview	853	entscheiden	903	entschieden	953	unklar
754	darin	804	stimmen	854	müller	904	italien	954	politischen
755	politiker	805	berichtete	855	polizisten	905	mehrheit	955	startseite
756	fordert	806	ganzen	856	ezb	906	her	956	erzählt
757	fahrer	807	smartphone	857	sache	907	zusammenarbeit	957	kostet
758	hohe	808	suchen	858	genommen	908	fotos	958	anderes
759	app	809	passiert	859	hause	909	börse	959	schlecht
760	gestellt	810	krieg	860	weise	910	vermutlich	960	brüssel
761	bekommt	811	news	861	überblick	911	sprach	961	geräte
762	morgen	812	ließ	862	verein	912	kollegen	962	großer
763	sprechen	813	chef	863	grenzen	913	geschäft	963	parlament
764	ihres	814	laufen	864	längst	914	vw	964	starken
765	seiten	815	zwölf	865	firma	915	verschiedenen	965	liebe
766	serie	816	wann	866	software	916	zusätzlich	966	euch
767	sprecher	817	putin	867	musik	917	häufig	967	mag
768	all	818	bundesregierung	868	führung	918	ähnlich	968	dritte
769	müssten	819	vater	869	komplett	919	boden	969	vergangenheit
770	unterwegs	820	größte	870	vorjahr	920	formel	970	raum
771	gefahr	821	somit	871	chancen	921	schön	971	stunde
772	gehören	822	wollten	872	christian	922	mehreren	972	tat
773	ergebnisse	823	mussten	873	gespräch	923	verhindern	973	herr
774	zürich	824	verlassen	874	drucken	924	verfahren	974	test
775	form	825	tod	875	persönliche	925	anderer	975	punkt
776	nein	826	erwarten	876	weder	926	zeiten	976	elf
777	film	827	union	877	lieber	927	beginnt	977	amazon
778	angekündigt	828	zeigte	878	kaufen	928	rede	978	forscher
779	monat	829	aktualisiert	879	funktioniert	929	leistung	979	suche
780	meist	830	umsatz	880	griechischen	930	setzte	980	fiel
781	nachricht	831	aktie	881	stellte	931	internationale	981	netz
782	immerhin	832	insbesondere	882	dinge	932	android	982	jedenfalls
783	sekunden	833	solchen	883	erfolgreich	933	schneller	983	nahe
784	athen	834	betroffen	884	verkauft	934	griechische	984	betonte
785	kind	835	gerät	885	videos	935	staatsanwaltschaft	985	partie
786	themen	836	bedeutet	886	gegner	936	glück	986	hintergrund
787	hieß	837	stärker	887	gewonnen	937	bundesliga	987	russische
788	york	838	jemand	888	glauben	938	rechnen	988	bisschen
789	trifft	839	erster	889	keiner	939	kämpfen	989	gewalt
790	kilometer	840	wolle	890	politische	940	getötet	990	spiegel
791	durchaus	841	bzw	891	verhandlungen	941	unterstützen	991	andreas
792	brauchen	842	stuttgart	892	preise	942	soldaten	992	plötzlich
793	hälfte	843	idee	893	fallen	943	empfehlen	993	egal
794	besteht	844	grünen	894	entwickler	944	fand	994	reihe
795	konzern	845	köln	895	vorbei	945	gezeigt	995	essen
796	st	846	wären	896	arbeitet	946	wahrscheinlich	996	schluss
797	ziehen	847	ändern	897	wenigen	947	dürften	997	produkte
798	los	848	russischen	898	meiner	948	kontrolle	998	manchmal
799	täter	849	ausland	899	mitteilte	949	geführt	999	gehe

1000 – 1050		1050 – 1100		1100 – 1150		1150 – 1200		1200 – 1250	
1000	verantwortlich	1050	dax	1100	focus	1150	job	1200	ungarn
1001	gewinn	1051	deutschlands	1101	patienten	1151	maßnahmen	1201	wechsel
1002	regeln	1052	manche	1102	buch	1152	entsprechend	1202	begonnen
1003	ermittlungen	1053	kindern	1103	schaden	1153	befindet	1203	mädchen
1004	schule	1054	glaube	1104	ots	1154	konkurrenz	1204	kennen
1005	luft	1055	gutes	1105	code	1155	möglicherweise	1205	red
1006	handel	1056	sorgt	1106	nahm	1156	manager	1206	dr
1007	international	1057	selber	1107	erkennen	1157	frankfurter	1207	technik
1008	rang	1058	tragen	1108	deutscher	1158	diesmal	1208	osten
1009	liga	1059	droht	1109	tut	1159	verstehen	1209	minus
1010	unfall	1060	sucht	1110	brachte	1160	zusammenhang	1210	vergessen
1011	london	1061	aufgabe	1111	hinweise	1161	links	1211	mercedes
1012	quelle	1062	dortmund	1112	möglichkeiten	1162	eins	1212	hoffnung
1013	wichtigsten	1063	nähe	1113	zeichen	1163	strecke	1213	langen
1014	obama	1064	tv	1114	fußball	1164	ums	1214	waffen
1015	anleger	1065	gesetz	1115	werbung	1165	liste	1215	getroffen
1016	zieht	1066	augen	1116	marke	1166	hilft	1216	informiert
1017	übrigens	1067	alt	1117	steigt	1167	ehemaligen	1217	betroffenen
1018	zeitpunkt	1068	hauptstadt	1118	grad	1168	startet	1218	präsentiert
1019	ziemlich	1069	anteil	1119	hamburger	1169	bessere	1219	falsch
1020	gäste	1070	französischen	1120	schutz	1170	führte	1220	amt
1021	unserem	1071	ball	1121	daniel	1171	reden	1221	hamilton
1022	anschließend	1072	wolfgang	1122	one	1172	fahrzeuge	1222	aufgenommen
1023	hinzu	1073	spitze	1123	gefallen	1173	täglich	1223	ernst
1024	kontakt	1074	pc	1124	europas	1174	erfahren	1224	freuen
1025	gleichen	1075	position	1125	verantwortung	1175	wobei	1225	risiko
1026	angela	1076	wort	1126	sogenannten	1176	wm	1226	aussagen
1027	wachstum	1077	steigen	1127	gold	1177	person	1227	verkauf
1028	stets	1078	apps	1128	stieg	1178	einzelnen	1228	überall
1029	müsste	1079	erhält	1129	ohnehin	1179	möglichst	1229	diskussion
1030	starke	1080	nötig	1130	gelten	1180	laufenden	1230	co
1031	westen	1081	zufrieden	1131	stück	1181	drittel	1231	sohn
1032	bahn	1082	tochter	1132	eingesetzt	1182	besucher	1232	gespräche
1033	com	1083	mitglieder	1133	offiziell	1183	basis	1233	nachfrage
1034	firmen	1084	wählen	1134	zuschauer	1184	investoren	1234	fällen
1035	verschiedene	1085	gewann	1135	genutzt	1185	lernen	1235	linie
1036	zugleich	1086	wohnung	1136	hannover	1186	gesamte	1236	iphone
1037	voll	1087	bern	1137	gespielt	1187	urteil	1237	verlieren
1038	rechte	1088	krise	1138	öffentlichen	1188	leisten	1238	legt
1039	moskau	1089	nummer	1139	angriff	1189	weiteres	1239	angeboten
1040	partner	1090	übernehmen	1140	kommentieren	1190	kostenlos	1240	mögliche
1041	ehemalige	1091	steuern	1141	reicht	1191	selten	1241	unten
1042	wiener	1092	smartphones	1142	iran	1192	langsam	1242	post
1043	entfernt	1093	herbst	1143	schweren	1193	gesetzt	1243	star
1044	streit	1094	kurs	1144	linken	1194	www	1244	starten
1045	leipzig	1095	bund	1145	stefan	1195	bürgermeister	1245	positiv
1046	heraus	1096	mensch	1146	genauso	1196	deswegen	1246	update
1047	berichten	1097	überzeugt	1147	meint	1197	tipp	1247	gebäude
1048	teilweise	1098	denken	1148	weiterer	1198	jüngsten	1248	prozess
1049	karriere	1099	solle	1149	höher	1199	finde	1249	training

1250 – 1300		1300 – 1350		1350 – 1400		1400 – 1450		1450 – 1500	
1250	gabriel	1300	grossen	1350	heutigen	1400	letztlich	1450	richter
1251	denke	1301	gefahren	1351	irak	1401	ministerpräsident	1451	aussage
1252	bisherigen	1302	früh	1352	anbieter	1402	bad	1452	zugang
1253	überraschend	1303	verhalten	1353	zug	1403	beendet	1453	xbox
1254	kanzlerin	1304	zusätzliche	1354	wolfsburg	1404	zählt	1454	publikum
1255	sah	1305	hören	1355	unternehmens	1405	großbritannien	1455	veröffentlichung
1256	qualität	1306	erhöht	1356	glaubt	1406	weltmeister	1456	eindruck
1257	kennt	1307	verfügbar	1357	erscheinen	1407	gestiegen	1457	sinn
1258	grosse	1308	mrd	1358	regel	1408	verbindung	1458	guter
1259	armee	1309	gesamten	1359	verwendet	1409	verzichten	1459	ziele
1260	extrem	1310	aktion	1360	zuerst	1410	notwendig	1460	gefordert
1261	aufs	1311	französische	1361	tiere	1411	beamten	1461	legen
1262	stattdessen	1312	düsseldorf	1362	niveau	1412	gesicht	1462	sachen
1263	wichtiger	1313	vertrauen	1363	verlor	1413	blatter	1463	werte
1264	genannt	1314	rechten	1364	freunde	1414	erinnert	1464	spielte
1265	spanien	1315	daraus	1365	sitzt	1415	zinsen	1465	nachfolger
1266	grundsätzlich	1316	gegangen	1366	entdeckt	1416	fuhr	1466	heimat
1267	parteien	1317	menge	1367	gern	1417	offensichtlich	1467	vorstellen
1268	beitrag	1318	washington	1368	zog	1418	niederlage	1468	entgegen
1269	alexander	1319	tore	1369	feuerwehr	1419	präsidenten	1469	bewusst
1270	stimmung	1320	wagen	1370	abschluss	1420	folgt	1470	polen
1271	forderte	1321	betrieb	1371	teilnehmer	1421	schmidt	1471	telekom
1272	aktiv	1322	rechts	1372	vorne	1422	sebastian	1472	champions
1273	kauf	1323	erklären	1373	gewählt	1423	gefragt	1473	britische
1274	auftritt	1324	öffentlichkeit	1374	einigung	1424	höhere	1474	kürzlich
1275	vorher	1325	erscheint	1375	bühne	1425	beiträge	1475	bedingungen
1276	früheren	1326	finale	1376	geschlossen	1426	bmw	1476	schützen
1277	mittel	1327	lebt	1377	gemeinde	1427	feuer	1477	fehlen
1278	gewinnt	1328	borussia	1378	familien	1428	rechnet	1478	steckt
1279	schwere	1329	richtige	1379	sitzen	1429	sehe	1479	spaß
1280	angeblich	1330	vorerst	1380	entsprechende	1430	energie	1480	gültige
1281	indem	1331	kritisiert	1381	name	1431	pläne	1481	beginnen
1282	samsung	1332	verbessern	1382	entwickeln	1432	relativ	1482	umfrage
1283	erlaubt	1333	gründen	1383	schwierig	1433	universität	1483	pause
1284	griechen	1334	david	1384	beteiligt	1434	jene	1484	bezeichnet
1285	legte	1335	sozialen	1385	schüler	1435	schauen	1485	gedacht
1286	möglichen	1336	tausende	1386	mobile	1436	halt	1486	entstehen
1287	wiederum	1337	großes	1387	finanzminister	1437	kamera	1487	japan
1288	betont	1338	bezahlen	1388	profitieren	1438	bedeutung	1488	traf
1289	bestätigte	1339	gefühl	1389	debatte	1439	verliert	1489	volk
1290	trägt	1340	handeln	1390	sid	1440	offiziellen	1490	vergrößern
1291	krankenhaus	1341	wirkt	1391	festgenommen	1441	fahrzeug	1491	geschrieben
1292	fifa	1342	alleine	1392	zentralbank	1442	generation	1492	erhältlich
1293	sogenannte	1343	journalisten	1393	verdient	1443	teile	1493	kleiner
1294	barcelona	1344	frühere	1394	wichtigen	1444	daraufhin	1494	angebote
1295	live	1345	nennt	1395	vierten	1445	alternative	1495	rosberg
1296	fährt	1346	licht	1396	besuch	1446	meinte	1496	israel
1297	erfahrung	1347	tritt	1397	plan	1447	winter	1497	produktion
1298	unbedingt	1348	gingen	1398	freien	1448	fordern	1498	koalition
1299	vorgestellt	1349	beschäftigt	1399	real	1449	organisation	1499	hielt

1500 – 1550		1550 – 1600		1600 – 1650		1650 – 1700		1700 – 1750	
1500	schuld	1550	davor	1600	grüne	1650	getan	1700	text
1501	hart	1551	eur	1601	rücken	1651	freiheit	1701	passieren
1502	markus	1552	liefern	1602	eröffnet	1652	übernommen	1702	presse
1503	modell	1553	schäuble	1603	meister	1653	reagiert	1703	verbunden
1504	ermöglicht	1554	automatisch	1604	verfügt	1654	klingt	1704	ursprünglich
1505	hoffen	1555	holen	1605	flüchtlingskrise	1655	jürgen	1705	vielmehr
1506	erwartungen	1556	schulden	1606	einzelne	1656	funktionen	1706	tot
1507	münchner	1557	wetter	1607	ferrari	1657	größer	1707	solange
1508	vertreter	1558	geraten	1608	mitteilung	1658	senden	1708	hotel
1509	gehabt	1559	linke	1609	ärzte	1659	ermittelt	1709	nahezu
1510	via	1560	terroristen	1610	schlägt	1660	beträgt	1710	auftrag
1511	verkaufen	1561	hervor	1611	feiern	1661	interessiert	1711	klare
1512	irgendwann	1562	jährlich	1612	salzburg	1662	gesprochen	1712	feld
1513	einst	1563	fernsehen	1613	audi	1663	gemeinsamen	1713	welchen
1514	kursziel	1564	schloss	1614	standen	1664	gleiche	1714	ehe
1515	nachrichtenagentur	1565	gelang	1615	bremen	1665	körper	1715	brand
1516	nachmittag	1566	öffentlich	1616	flughafen	1666	bestimmt	1716	wunsch
1517	geplanten	1567	stimmt	1617	vettel	1667	stimme	1717	raus
1518	inhalte	1568	toten	1618	pegida	1668	kunst	1718	guardiola
1519	meinem	1569	konzept	1619	erstes	1669	plant	1719	integration
1520	begann	1570	dienst	1620	umsetzung	1670	lösungen	1720	wettbewerb
1521	verändert	1571	strategie	1621	van	1671	lief	1721	angreifer
1522	verurteilt	1572	kirche	1622	reisen	1672	tages	1722	praktisch
1523	falls	1573	demokratie	1623	group	1673	zahlreichen	1723	nennen
1524	tief	1574	aufnehmen	1624	reichen	1674	zentrum	1724	analyst
1525	gesperrt	1575	bord	1625	erhielt	1675	top	1725	piloten
1526	frühen	1576	schulen	1626	geschafft	1676	entspricht	1726	autor
1527	vorgehen	1577	volkswagen	1627	leiter	1677	rest	1727	erlebt
1528	gründe	1578	erde	1628	robert	1678	gruppen	1728	öffentliche
1529	befinden	1579	bewegung	1629	erhöhen	1679	ermöglichen	1729	redaktion
1530	abgeschlossen	1580	basel	1630	vertreten	1680	anklicken	1730	verpflichtet
1531	außer	1581	heisst	1631	bundestag	1681	mario	1731	super
1532	regelmäßig	1582	karte	1632	metern	1682	sorgte	1732	runden
1533	ansicht	1583	computer	1633	gezogen	1683	straßen	1733	matthias
1534	reaktionen	1584	private	1634	initiative	1684	frank	1734	management
1535	roten	1585	ca	1635	auswirkungen	1685	city	1735	nutzung
1536	richtigen	1586	kündigte	1636	bezahlt	1686	lösen	1736	gelungen
1537	investitionen	1587	sprache	1637	warnt	1687	freude	1737	posten
1538	welcher	1588	flucht	1638	service	1688	bauen	1738	freie
1539	welches	1589	anstieg	1639	schweden	1689	fälle	1739	geplante
1540	griechenlands	1590	entsprechenden	1640	schafft	1690	erzielte	1740	ansonsten
1541	anschlag	1591	deutsch	1641	fokus	1691	größere	1741	verstärkt
1542	schalke	1592	möchten	1642	mitglied	1692	satz	1742	afd
1543	passt	1593	einfluss	1643	erzielt	1693	regionen	1743	sanktionen
1544	britischen	1594	verletzungen	1644	rückkehr	1694	händler	1744	hinweis
1545	reise	1595	branche	1645	tisch	1695	bekam	1745	zählen
1546	john	1596	freund	1646	amerikanischen	1696	fanden	1746	mehrfach
1547	kultur	1597	sparen	1647	bestehen	1697	treffer	1747	verbessert
1548	generieren	1598	industrie	1648	gehalten	1698	fürs	1748	übernahme
1549	gmbh	1599	zweifel	1649	sony	1699	bau	1749	fühlen

1750 – 1800		1800 – 1850		1850 – 1900		1900 – 1950		1950 – 2000	
1750	million	1800	zürcher	1850	lufthansa	1900	öffnen	1950	ausschließlich
1751	option	1801	band	1851	coach	1901	umfeld	1951	hinten
1752	madrid	1802	bekannte	1852	worten	1902	ausgabe	1952	glücklich
1753	zeugen	1803	funktion	1853	terror	1903	wirft	1953	verteilt
1754	teuer	1804	paul	1854	download	1904	unbekannte	1954	klicken
1755	rein	1805	reaktion	1855	dich	1905	rücktritt	1955	österreichischen
1756	trend	1806	derweil	1856	hsv	1906	club	1956	verbraucher
1757	unterschied	1807	engagement	1857	gegensatz	1907	blieben	1957	umgesetzt
1758	chinesischen	1808	positive	1858	antrag	1908	veranstaltung	1958	verlangt
1759	privaten	1809	gemeinden	1859	teils	1909	weiss	1959	verfolgen
1760	machten	1810	aufgaben	1860	feiert	1910	unseres	1960	ios
1761	bayer	1811	dresden	1861	reagieren	1911	geschehen	1961	asylbewerber
1762	vorwürfe	1812	analyse	1862	beteiligten	1912	halle	1962	bundesrat
1763	wohnungen	1813	hoffe	1863	interessieren	1913	soziale	1963	tablet
1764	versuch	1814	games	1864	gaben	1914	staffel	1964	prüfen
1765	einführung	1815	einschätzung	1865	indes	1915	investieren	1965	vorteil
1766	ans	1816	erfolgt	1866	näher	1916	wichtigste	1966	verdienen
1767	jugendliche	1817	hans	1867	erfahrungen	1917	christoph	1967	leverkusen
1768	anspruch	1818	treten	1868	änderungen	1918	pressekonferenz	1968	plätze
1769	einiges	1819	tonnen	1869	mache	1919	verkehr	1969	jobs
1770	csu	1820	übernimmt	1870	vorhanden	1920	plattform	1970	maximal
1771	interessant	1821	kandidaten	1871	bedarf	1921	hängt	1971	kreis
1772	strom	1822	bewegen	1872	standard	1922	brasilien	1972	verhältnis
1773	herausforderung	1823	entweder	1873	technischen	1923	klasse	1973	jeweiligen
1774	worte	1824	erklärung	1874	gegenteil	1924	arbeitgeber	1974	weitgehend
1775	laufe	1825	erleben	1875	dringend	1925	hunderte	1975	gesucht
1776	seitdem	1826	behandelt	1876	verwenden	1926	hofft	1976	frühjahr
1777	schließen	1827	bekannten	1877	hundert	1927	nutzt	1977	mitarbeitern
1778	miteinander	1828	reuters	1878	gesundheit	1928	praxis	1978	chinesische
1779	ideen	1829	ps	1879	iwf	1929	digitale	1979	gebaut
1780	gestartet	1830	auswahl	1880	sonne	1930	handy	1980	retten
1781	stolz	1831	ausbildung	1881	enthalten	1931	verlust	1981	risiken
1782	geschäftsführer	1832	anzahl	1882	bestimmte	1932	ii	1982	vorgesehen
1783	ständig	1833	dar	1883	laden	1933	verband	1983	übrig
1784	szene	1834	gedanken	1884	gemeinsame	1934	reformen	1984	geblieben
1785	max	1835	bundeskanzlerin	1885	analysten	1935	sv	1985	varoufakis
1786	kanton	1836	eurozone	1886	dauern	1936	autofahrer	1986	standort
1787	umgang	1837	charlie	1887	tätig	1937	anschluss	1987	bewohner
1788	irgendwie	1838	politisch	1888	anfrage	1938	höheren	1988	dahin
1789	beschlossen	1839	russlands	1889	vfl	1939	entscheidungen	1989	natur
1790	leistungen	1840	zweimal	1890	stürmer	1940	eingestellt	1990	schlagzeilen
1791	zunehmend	1841	australien	1891	wächst	1941	duell	1991	künstler
1792	ermittler	1842	england	1892	fdp	1942	untersuchung	1992	hört
1793	außerhalb	1843	la	1893	pkw	1943	bvb	1993	schlagen
1794	projekte	1844	opposition	1894	ruhe	1944	momentan	1994	kommission
1795	persönlich	1845	benötigt	1895	gehandelt	1945	letztes	1995	wahlen
1796	le	1846	svp	1896	vorschlag	1946	berufung	1996	unabhängig
1797	tipps	1847	norden	1897	tote	1947	überrascht	1997	islamischer
1798	fahrt	1848	voraussichtlich	1898	modelle	1948	nannte	1998	forderungen
1799	besondere	1849	spö	1899	spätestens	1949	weshalb	1999	übertragen

PoS	Test Case	Lemma	Frequency Rank		
			≤500	≤1000	≤2000
Noun	Gender Singular/ Plural	male	Mann	Männer Junge Jungen	
		female	Frau Frauen		Mädchen
	Declination Singular/ Plural	day	Tag Tage Tagen		Tages
		year	Jahr Jahre Jahren Jahres		
		smartphone		Smartphone	Smartphones
		Land	Land	Landes Länder Ländern	
	Derivation	driver		Fahrer	Autofahrer
		player	Spieler		
		movement			Bewegung
		thought			Gedanken
		feeling		Gefühl	
		help		Hilfe	
Proper Noun	Entailment	Russia	Russland		Russlands
		Moscow			Moskau
		America	USA		
		Washington			Washington
		Europe	Europa		Europas
		England			England
		London			London
		France		Frankreich	
		Paris	Paris		
Verb	Conjugation Derivation	(to) take	nehmen	nimmt genommen	nahm
		(to) go	gehen geht ging	gehe	gingen gegangen
		(to) have	haben habe hat hatte hatten		gehabt
		(to) do	tun	tat	getan tut
		(to) play	spielen	spielt	spielte gespielt
		(to) show	zeigen zeigt	zeigte gezeigt	
		(to) make	machen macht gemacht	machte	machten mache
		(to) drive		fahren	gefahren
		(to) move			bewegen
		(to) think			denken denke gedacht
		(to) feel			fühlen
		(to) help		helfen	hilft
		(to) open			öffnen
		(to) receive	erhalten		erhielt erhält
Adjectives Adverbs	Declination Comparative Derivation Antonymy	high	große großen	groß großer größten größte	größer größere großes
		small		kleine kleinen	kleiner
		good	gut gute besser besten	guten beste	gutes guter bessere
		bad		schlecht	
		American			amerikanischen
		Russian	russische	russischen	
		European	europäischen	europäische	
		open			offen
		available			erhältlich
(Compound-) Noun	OOV	clock	Uhr		
		Federal Government		Bundesregierung	
		Managing Director			Geschäftsführer

TABLE B.1: List of German Words for the Evaluation

Bibliography

Adams, D. (1979). *The Hitchhiker's Guide to the Galaxy*. Chapter 6.

Agirre, E. and Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics.

Alexandrescu, A. and Kirchhoff, K. (2006). Factored neural language models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 1–4. Association for Computational Linguistics.

Anderson, S. R. (2010). How many languages are there in the world. *Linguistic Society of America*.

Arnold, D., Balkan, L., Meijer, S., Humphreys, R., and Sadler, L. (1994). Machine translation: An introductory guide. NCC Blackwell.

Artetxe, M., Labaka, G., and Agirre, E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294.

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 451–462.

Ayan, N. F., Dorr, B. J., and Habash, N. (2004). Multi-Align: Combining linguistic and statistical techniques to improve alignments for adaptable MT. In *Conference of the Association for Machine Translation in the Americas*, pages 17–26. Springer.

Bach, F. R. and Jordan, M. I. (2005). A probabilistic interpretation of canonical correlation analysis.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247.

Beesley, K. R. and Karttunen, L. (2003). Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*.

Bengio, Y., Ducharme, R., and Vincent, P. (2001). A neural probabilistic language model. In *Advances in Neural Information Processing Systems*, pages 932–938.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Berardi, G., Esuli, A., and Marcheggiani, D. (2015). Word embeddings go to italy: A comparison of models and training datasets. In *IIR*.

Besson, L. and Kamen, R. M. (1997). The Fifth Element.
Movie, produced by Patrice Ledoux, and published by Columbia Pictures.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational linguistics*, 16(2).

Brown, P. F., Pietra, V. J. D., Mercer, R. L., Pietra, S. A. D., and Lai, J. C. (1992). An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40.

Bullinaria, J. A. and Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. (2017). Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 854–863. JMLR. org.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Daciuk, J., Mihov, S., Watson, B. W., and Watson, R. E. (2000). Incremental construction of minimal acyclic finite-state automata. *Computational linguistics*, 26(1):3–16.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Dorow, B., Laws, F., Michelbacher, L., Scheible, C., and Utt, J. (2009). A graph-theoretic algorithm for automatic extension of translation lexicons. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 91–95.

Dryer, M. S. and Haspelmath, M., editors (2013). *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. 1952-59:1–32.

Franz, A., Horiguchi, K., Duan, L., Ecker, D., Koontz, E., and Uchida, K. (2000). An integrated architecture for example-based machine translation. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*, volume 2.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule. Technical report, COLUMBIA UNIV NEW YORK DEPT OF COMPUTER SCIENCE.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *LREC*, volume 29, pages 31–43.

Golub, G. H. and Loan, C. (2013). Matrix computations, forth edition.

Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goodfellow, I., Pouget Abadie, J., Mirza, M., Xu, B., Warde Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gough, N. and Way, A. (2004). Example-based controlled translation.

Gurevych, I. (2005). Using the structure of a conceptual network in computing semantic relatedness. In *International Conference on Natural Language Processing*, pages 767–778. Springer.

Gutmann, M. U. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(Feb):307–361.

Haghighi, A., Liang, P., Berg Kirkpatrick, T., and Klein, D. (2008). Learning bilingual lexicons from monolingual corpora. *Proceedings of ACL-08: Hlt*, pages 771–779.

Hamp, B. and Feldweg, H. (1997). Germanet-a lexical-semantic net for german. *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Harabagiu, S. M., Miller, G. A., and Moldovan, D. I. (1999). Wordnet 2-a morphologically and semantically enhanced resource. *SIGLEX99: Standardizing Lexical Resources*.

Hardoon, D. R., Szedmak, S., and Shawe Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hassan, S. and Mihalcea, R. (2009). Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1201.

Haveliwala, T. H. (2002). Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM.

Henrich, V., Hinrichs, E., and Vodolazova, T. (2011). Semi-automatic extension of GermaNet with sense definitions from Wiktionary. In *Proceedings of the 5th Language and Technology Conference (LTC 2011)*, pages 126–130.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2001). Introduction to automata theory, languages, and computation. Second Edition.

Hotelling, H. (1936). Relations between two Sets of Variates. *Biometrika*, 28(3/4):321–377.

Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101.

Huxley, T. H. (1870). *Address to the British Association for the Advancement of Science*. Taylor. Page 11.

Jeh, G. and Widom, J. (2002). SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM.

Jelinek, F., Mercer, R. L., Bahl, L. R., and Baker, J. K. (1977). Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1.

Joubarne, C. and Inkpen, D. (2011). Comparison of semantic similarity for different languages using the google n-gram corpus and second-order co-occurrence measures. In *Canadian Conference on Artificial Intelligence*, pages 216–221. Springer.

Kalman, D. (2009). Leveling with lagrange: An alternate view of constrained optimization. *Mathematics Magazine*, 82(3):186–196.

Kaplan, R. M. and Kay, M. (1994). Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Karlgren, J. and Sahlgren, M. (2001). From words to understanding. *Foundations of real-world intelligence*, pages 294–308.

Koehn, P. (2017). Neural machine translation. *arXiv preprint arXiv:1709.07809*.

Koehn, P. and Knight, K. (2002). Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*.

Köper, M., Scheible, C., and im Walde, S. S. (2015). Multilingual reliability and "semantic" structure of continuous word spaces. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 40–45.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Kunze, C. and Lemnitzer, L. (2002). GermaNet-representation, visualization, application. In *LREC*.

Landauer, T. K. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Laws, F., Michelbacher, L., Dorow, B., Scheible, C., Heid, U., and Schütze, H. (2010). A linguistically grounded graph model for bilingual lexicon extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 614–622. Association for Computational Linguistics.

Lazaridou, A., Marelli, M., Zamparelli, R., and Baroni, M. (2013). Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1517–1526.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Lin, C.-Y. and Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612.

Lovász, L. et al. (1993). Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, 2(1):1–46.

Luong, T., Socher, R., and Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Mihalcea, R. (2004). Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*.

Mikolov, T. (2012). Statistical language models based on neural networks. *PhD thesis, Brno University of Technology*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, T., Sutskever, I., Deoras, A., Le, H.-S., Kombrink, S., and Cernocky, J. (2012). Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*, 8.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Miller, G. A. (1995). *Communications of the ACM*, 38(11):39–41.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Minkov, E. and Cohen, W. (2012). Graph based similarity measures for synonym extraction from parsed text. In *Workshop Proceedings of TextGraphs-7: Graph-based Methods for Natural Language Processing*, pages 20–24.

Niehues, J. and Waibel, A. (2012). Detailed analysis of different strategies for phrase table adaptation in SMT. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*.

Noveck, I. A. and Sperber, D. (2004). *Experimental pragmatics*. Springer.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 295–302. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational linguistics*, 30(4):417–449.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Panchenko, A., Ustalov, D., Arefyev, N., Paperno, D., Konstantinova, N., Loukachevitch, N., and Biemann, C. (2016). Human and machine judgements for russian semantic relatedness. In *International conference on analysis of images, social networks and texts*, pages 221–235. Springer.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pillai, S. U., Suel, T., and Cha, S. (2005). The Perron-Frobenius theorem: some of its applications. *IEEE Signal Processing Magazine*, 22(2):62–75.

Rapp, R. (1995). Identifying word translations in non-parallel texts. *arXiv preprint cmp-lg/9505037*.

Rapp, R. (1999). Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 519–526. Association for Computational Linguistics.

Reynolds, C. R. (1983). Test bias: In God we trust; all others must have data. *The Journal of Special Education*, 17(3):241–260.

Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Rothe, S. and Schütze, H. (2014). Cosimrank: A flexible & efficient graph-theoretic similarity measure. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1392–1402.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

Ruppenhofer, J., Ellsworth, M., Schwarzer Petruck, M., Johnson, C. R., and Scheffczyk, J. (2006). FrameNet II: Extended theory and practice.

Sahlgren, M. (2005). An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering*.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Schütze, H. (1993). Word space. In *Advances in neural information processing systems*, pages 895–902.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

Smith, S. L., Turban, D. H., Hamblin, S., and Hammerla, N. Y. (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Svoboda, L. and Brychcin, T. (2016). New word analogy corpus for exploring embeddings of czech words. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 103–114. Springer.

Thurmair, G. (2005). Hybrid architectures for machine translation systems. *Language Resources and Evaluation*, 39(1):91–108.

Tukey, J. W. (1977). *Exploratory data analysis*, volume 2. Reading, MA.

Turney, P. D. (2006). Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Uurtio, V., Monteiro, J. M., Kandola, J., Shawe Taylor, J., Fernandez Reyes, D., and Rousu, J. (2018). A tutorial on canonical correlation methods. *ACM Computing Surveys (CSUR)*, 50(6):95.

Voorhees, E. M. and Tice, D. M. (1999). The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82. Citeseer.

Vossen, P. (2002). EuroWordNet: general document.

Weaver, W. (1955). Translation. *Machine translation of languages*, 14:15–23.

Wittgenstein, L. (1953). Philosophical investigations. Philosophische Untersuchungen.

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403.

Zar, J. H. (2005). Spearman rank correlation. *Encyclopedia of Biostatistics*, 7.

Zesch, T. and Gurevych, I. (2006). Automatically creating datasets for measures of semantic relatedness. In *Proceedings of the Workshop on Linguistic Distances*, pages 16–24. Association for Computational Linguistics.

Zimmermann, T. E. and Sternefeld, W. (2013). *Introduction to semantics: An essential guide to the composition of meaning*. Walter de Gruyter.

Zou, W. Y., Socher, R., Cer, D., and Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398.