Evaluating Bilingual Embeddings in Bilingual Dictionary Alignment

Çift Dilli Kelime Temsilleri ile Sözlük Eşlenmesi

Yiğit Sever

Dr. Gönenç Ercan

Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfilment to the Requirements
for the Award of the Degree of Master of Science
in Computer Engineering

2019

Declaration of Authorship

I, Yiğit Sever, declare that this thesis titled, "Evaluating Bilingual Embeddings in Bilingual Dictionary Alignment" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

Abstract

Yiğit Sever

Evaluating Bilingual Embeddings in Bilingual Dictionary Alignment

Dictionaries catalog and describe the semantic information of a lexicon. WordNet provides an edge by presenting distinct concepts with the hierarchy information among them. Research in computer science has been using this hand crafted tool in natural language applications such as text summarization and machine translation. Original WordNet has been compiled for English yet counterparts for other languages are not as readily available nor as comprehensive. In order for research on languages other than English to benefit from the power of a WordNet, machine assisted creation and evaluation methods are essential.

Word embeddings can provide a mapping between words and points in a real valued vector space. Using these vectors, representing documents as well as forming geometric relationships between them is a well studied area of research. In this thesis we start by hypothesizing that a dictionary definition captures the semantic basis of the described word. We used word embeddings as building blocks to map dictionary definitions into a multidimensional space. These spaces can be aligned to accommodate two languages, allowing the transfer of information from one language to another. We investigate the success of retrieving and matching discrete senses across languages by employing supervised and unsupervised methods. Our experiments show that dictionary alignment can be evaluated successfully by using both unsupervised and supervised methods but corpora sizes should be taken into consideration. We further argue that some methods are not viable considering their poor performance.

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Contents

List of Figures

List of Tables

Aileme...

1. Introduction

1.1. Dictionaries

Dictionaries are living records of a society's language usage. Languages change over time, people adopt new words for new senses while others fall out of use. Concepts appear as a result of technological advancements or social shifts, giving birth to new senses and words to define them. Meanwhile, the term *dictionary* is a broad one to define. On its own, it brings forth the monolingual dictionary into consideration [1]. This type of dictionary presents words alongside their definitions following an alphabetical order. The intention is to inform the user about the words [2]. Other types of dictionaries vary with regard to their use case, target audience, and scope. For instance, bilingual dictionaries present words alongside their translations in the target language, often used by language learners or translators. Domain specific dictionaries list technical terms that target people who are familiar with the terminology.

The term that precedes the entries is called *headword* or *lemma*. Usually, lemmas are the form of a word without inflections. The sense they convey is as comprehensive as possible, reducing the number of otherwise redundant entries that would have been the derivatives of the unmarked form [3].

Dictionaries also inform the user about how senses relate to each other. **Polysemous** words share the same spelling while having related, often derivative meanings. For example; under the entry for the term *bank*, a definition might clarify the meaning *financial institution* while another can define *the building of a financial institution*. In contrast, **homonymous** words have distinct meanings while having identical spellings through coincidence. Formal definition of homonymy separates sound based and spelling based homonymy differently as homophones and homographs but for the purposes of our text based arguments, we do not delve into the specifics. The *bank of a river* is homonym to the given examples. Homonyms are often shown in discrete blocks of descriptions.

Synonymity is another lexical relation we are interested in. A word is synonymous to another if they share the same meaning but are not spelled alike, such as the terms *right* and *correct*. However, synonymity is seldom shown in dictionaries.

Dictionaries take an immense amount of time and expertise to prepare. We can talk about the examples after narrowing our scope down to the dictionaries that are still

available today. A survey by Uzun [4] notes that the first instalment of the modern Turkish dictionary, led by a team of experts, has taken over 6 years to prepare. Kendall [5] talks about how Noah Webster, the writer of the *An American Dictionary of the English Language* had to mortgage off his home in order to finish his project which took over 26 years. The bulk of this effort is collecting documents and other written material in order to establish a *corpus* [4]. This endeavour is necessary since a corpus is crucial to create the vocabulary of a language. Once the corpus is at hand, researchers can extract the lemmas. The resulting wordstock is called the *lexicon* of the language.

The internet radically changed the way researchers aggregate data. The advancements in digital storage technology allowed the data to be persistent. Improvements in networking ensured that people can share the volume of it among themselves. With the popularization of social media, the internet generates everyday conversations at an unprecedented rate that researchers are using for natural language applications. Moreover, efforts on open, collaborative, web based encyclopedias generate structured, multilingual data often used in machine translation and text categorization tasks. Once the cumbersome task of corpus attainment is now akin to web crawling. With the digitized data, it was only natural for dictionaries to go digital as well since it's generally acknowledged that they are no longer viable if they are not electronic [1].

1.2. WordNet

George A. Miller started the WordNet project in the mid-1980s. On its early days, project members studied theories that were aimed towards enabling computers to understand natural language as intrinsically as humans do. While working on then popular semantic networks and sense graphs, they have started something that will evolve into an expansive, influential resource [6].

Traditional dictionaries are rigid, constrained by the nature of the printed form. Today, people can browse WordNet via queries, like an online dictionary or a thesaurus. Behind the scenes, a sprawling lexical database has relationship information for more than 117000 senses. Figure 1.1 shows a brief result for the query string "run".

WordNet lists terms, much like a traditional dictionary, alongside its polysemes but also their homonyms. Additionally, there is a horizontal association; for any sense, the lemmas that share the row with the target term are synonyms. Furthermore, synset terms can have one or more lexemes, not necessarily singular tokens. This set

Noun

S: (n) **run**, tally (a score in baseball made by a runner touching all four bases safely) *"the Yankees scored 3 runs in the bottom of the 9th"; "their first tally came in the 3rd inning"*

direct hyponym / full hyponym

- S: (n) earned run (a run that was not scored as the result of an error by the other team)
- S: (n) unearned run (a run that was scored as a result of an error by the other team)
- S: (n) run batted in, rbi (a run that is the result of the batter's performance) *"he had more than 100 rbi last season"*

direct hypernym / inherited hypernym / sister term

- S: (n) score (the act of scoring in a game or sport) *"the winning score came with less than a minute left to play"*

derivationally related form

- W: (v) run [Related to: run] (make without a miss)
- W: (v) tally [Related to: tally] (keep score, as in games)
- W: (v) tally [Related to: tally] (gain points in a game) *"The home team scored many times"; "He hit a home run"; "He hit .300 in the past season"*

S: (n) test, trial, **run** (the act of testing something) *"in the experimental trials the amount of carbon was measured separately"; "he called each flip of the coin a new trial"*

S: (n) footrace, foot race, **run** (a race run on foot) *"she broke the record for the half-mile run"*

Verb

- S: (v) **run** (move fast by using one's feet, with one foot off the ground at any given time) *"Don't run--you'll be out of breath"; "The children ran to the store"*
- S: (v) scat, **run**, scarper, turn tail, lam, run away, hightail it, bunk, head for the hills, take to the woods, escape, fly the coop, break away (flee; take to one's heels; cut and run) *"If you see this man, run!"; "The burglars escaped before the police showed up"*

FIGURE 1.1: WordNet result for the query "run", truncated for brevity.

of synonyms is aptly named *synsets*. An example synset is {*ledger*, *account book* and *book of account*}.

A short description is also provided to clarify the meaning. For the synset given above, the definition is; "a record in which commercial accounts are recorded" These descriptions, hence the meanings for any synset is unique within the WordNet. During this discussion, we have used sense and synset interchangeably.

WordNet also includes other relationships such as *hypernymy* and *hyponymy*, semantic relation of senses being type-of one another [7].[1] For instance, the term "building" is a hyponym of "restaurant" since it encompasses a more general sense; the restaurant is type of a building. While coffee shop is a hypernym to the restaurant since it is a more specific sense.

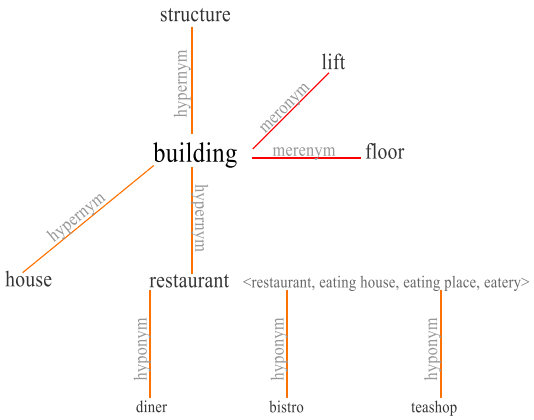

FIGURE 1.2: Example of a WordNet relationship graph. Hyponymy/hypernymy relations are shown from the hierarchy level of *restaurant*

One other relation is the meronymy, defined as a sense being part of or a member of another [8]. Keeping to our building example, *windows* are *meronym* to buildings. Other relationships are also available but the bottom line is the effort that has gone through to map 117,000 senses according to different semantic relationships.

[1] not to be confused with homonymy

sagot_building_2008 argue that the semantic relationships between senses are not tied to a specific language. In other words, preparing new wordnets via extending the English Princeton WordNet ultimately works because semantic relationships between senses are language invariant. With this information at hand, we can infer the effort behind the WordNet does not need to be fully repeated for other languages.

Since it's inception, other projects built lexical databases, using the same WordNet design. Fellbaum [6] talks about the correct terminology that we abide for the thesis; "As WordNet became synonymous with a particular kind of lexicon design, the proper name shed its capital letters and became a common designator for semantic networks of natural languages". Hence *WordNet* refers to English Princeton WordNet, while *wordnets* created for other languages are not stylized.

1.3. Multilingual Wordnets

Authorities list more than 7000[2] living languages but only 40[3] of them have a sizeable presence on the internet. Among this small fraction, English is the dominant language of the web. English in not the centrepiece for natural language processing research because of any linguistic attribute. It is simply the most abundant language on web, giving researchers data to work with.

Natural language processing library spaCy[4] resorts to lemmatizations such as $=PRON=$ to denote pronouns in order to collapse the senses for "I" "you", "them" etc.. The sense and the accompanying word for the brother of a person's father or mother differs in Turkish, Danish, Chinese and Swedish among other languages while both collapse to "uncle" in English. Studying other languages can provide insight towards concepts that are not present in English.

Translation, information transfer from foreign languages is a valid way of enriching a language's corpora; if a term that for a sense does not have a match in the target language, it is a good indication for the linguists of that language to look into their lexicons and work towards expanding it [3]. Further research in the area contributes to languages other than English having access to tools that will incorporate them into the literature.

[2] https://www.ethnologue.com/statistics
[3] https://w3techs.com/technologies/history_overview/content_language
[4] https://spacy.io

Open Multilingual WordNet [9, 10] set out to discover the effects related to the choice of license for wordnets. Their criteria for usefulness is the number of citations a publication tied to the wordnet has gotten on literature. They identified two major problems with the current distributions;

- some projects have picked restrictive licenses, effectively barring access to their tools for research purposes.
- the structures of the wordnets are not standardized, creating additional cost for creating programs to parse and use the wordnets.

In order to overcome the standardization issue, Bond & Paik have aligned the wordnets according to their English Princeton WordNet lemma ids and have written individual scripts to parse them. They are currently hosting the results from a single source.[5]

With alignment information at hand, we have created our dataset that we will assume to be perfectly aligned; a golden corpus. Among the 34 wordnets available on Open Multilingual WordNet, only 6 of them have gloss information available. Given this thesis will only investigate the ability to map senses using definitions of the sense, we used the subset of Albanian [11], Bulgarian [12], Greek [13], Italian [14], Slovenian [15] and Romanian [16] wordnets. Table 1.1 shows brief statistics about them. We should note that the languages of the wordnets used in the thesis are all present in the 40 languages that have a significant presence on the internet that we have mentioned before. We have constrained this study to use only the freely available wordnets and not considered wordnets that are gated behind restrictive licenses.

Name of the Project	**Language**	**Number of Definitions**
Albanet	Albanian	4681
BulTreeBank WordNet	Bulgarian	4959
Greek Wordnet	Greek	18136
ItalWordnet	Italian	12688
Romanian Wordnet	Romanian	58754
SloWNet	Slovenian	3144

TABLE 1.1: Summary of the Wordnets used.

[5] http://compling.hss.ntu.edu.sg/omw

1.4. Thesis Goals

In this thesis, we will study the dictionary alignment problem. It naturally arises on wordnet generation tasks when extend approach is used. Wordnet generation as well as extend approach will be talked about in detail in Section 2.2. It can be used in conjunction with word sense disambiguation frameworks to match the correct sense of a term across languages.

Two unsupervised approaches will be presented as an answer to dictionary alignment problem. First one is the matching approach, in which the dictionary definitions across languages will be thought as nodes of a weighted bipartite graph. By assigning appropriate weights to the edges with respect to distance between individual dictionary definitions, we hypothesize that there is a case where minimum flow is achieved between the disjoint dictionary sets. The same case can also be stated as the case where the edge weights of the matching is maximum if a similarity metric is used to assign weights. The second approach will be a unsupervised document retrieval approach, adapted on dictionary definitions. An algorithm that works with the consideration of individual distances between words will be presented.

We will also look into the supervised sentence alignment using a neural network approach. An encoder will be trained on a metric for determining if two sentences that were written in different languages entail the same sense. The performance of the encoder will hopefully give us insight towards using a bag of words model of sentence representation, decoupled from syntax of the language can be viable for the task.

All in all, we will investigate the following research questions;

- How feasible is it to align senses across languages using their dictionary definitions.
- Which state of the art algorithms is most suitable for the task.
- Which parameters should be taken into consideration while tackling this task

The comparative study will be done using existing word embedding models. This highlights the major advantage of our approach; current word embedding models are trained on billions of tokens to learn the distributional properties of the words. Comparatively, dictionaries are short texts. However, we can bring the word embeddings trained out of domain to solve in domain tasks.

1.5. Thesis Outline

First, we present the most important preliminary to our study in Chapter 2, the word embeddings. Word embeddings provided the crucial basis for out thesis. Without relying on representing words in a multidimensional space, establishing distance formulation simply would not work. So we report on the history of how this representations came about, present the current popular approaches to learning and distributing word embeddings and discuss the models we have chosen in detail. Even though the presented approaches can work with any two pairs of dictionary definition collections, one practical use for this application using the alignment process to extend the semantic database WordNet. We report on approaches on wordnet generation using the previous works that created wordnets. Other related work on representing senses are briefly discussed on Chapter 2 as well.

In order to address our research questions, we have looked into approaches that can be broken down into 3 separate categories.

1. Matching approach
2. Retrieval approach
3. Supervised approach

Chapter 3 is concerned with the exploration of the unsupervised matching techniques. Here we will present how dictionary alignment task can be cast as a bipartite graph matching problem and report on our approach using linear programming.

We will handle dictionary alignment problem using state of the art document retrieval algorithms in Chapter 4. How the current approaches leverage word embeddings in order to represent words as probability distributions that lie on a multidimensional simplex will be presented in detail. Furthermore, we will explain the algorithms behind how efficient distance calculations can be achieved in this problem setting.

Chapter 5 is reserved for our supervised approach. Given two definitions collections that we accept as perfectly aligned, we will investigate if an encoder can learn the semantic similarity as a function given positive and negative examples. In order to present our problem, we will report on the current state of the art approach on semantic similarity and the neural network model that forms the basis of said approach.

Since our thesis is heavily concerned with investigating the best approach for the task at hand, we collected all our results in Chapter 6. By presenting all results from a single place, we hope to give a complete picture on how we have attacked the dictionary alignment task, which approaches performed better against others and the findings that emerged within the results. Details concerning our implementation, how we obtained the aligned corpora and the word embeddings that share the same latent space are explained in Chapter 6 as well. Finally, we will conclude our thesis in Chapter 7 with an overall look on our findings and the future work that we would like to study next.

2. Background Information & Related Work

Somers puts down modern dictionaries in their article *You're Probably Using the Wrong Dictionary*; "The definitions are these desiccated little husks of technocratic meaningese, as if a word were no more than its coordinates in semantic space." [17]. From the perspective of an author, the efficiency of dictionaries might be worrisome but we will build this thesis on the presumption that dictionary definitions can indeed be represented in some semantic space using the words they are written with as the elements of their vectors.

2.1. Word Embeddings

Word embeddings are real valued dense vectors that represent words. Recently, language modelling studies have been focusing on explicitly learning word embeddings in order to show words or phrases as points on a low dimensional latent space. On the other hand, earlier research had been obtaining what we can call feature vectors for words while studying natural language processing tasks such as named entity recognition or part of speech tagging [18, 19]. The vector representation allows researchers access to the tools of the broad literature in linear algebra and machine learning, since they are intuitive for humans to interpret, more so for machines. Vectors can be compared as a measure of semantic similarity or composed together to build more expansive sentence, paragraph or document representations.

Induced embeddings can be saved to the disk in matrix notation. Each row is labelled with a token which is shown in some space by the following n real numbers, as popularized by the open source package *word2vec*. Researchers have been sharing their models on the internet so that other researchers can simply download and use them in their own applications [**kusner_word_2015**, 20, 21]. Word embeddings available in this manner are often called as *pre-trained* models. Examples of pre-trained word embeddings are *word2vec* [22], *GloVe* [23], *Numberbatch* [21] and *fastText* [**bojanowski_enriching_2016**].

In the following section, we will briefly present the history of word embeddings. Research on word embeddings is a sprawling subject that researchers has been building upon using the ideas from probabilistic, statistical and neural network models. We

have omitted crucial contributions that optimized models and brought the literature where it is today due to space constraints, following a path that will lead into the preliminary behind the models we have chosen.

2.1.1. History of Word Representations

In order to talk about how words can be mapped to a multidimensional space, we should first talk about how the idea that *they can* has came about.

Linguistic Background

In his 1954 article, Harris [24] introduced his ideas which later came to known as *distributional hypothesis* in the field of linguistics. He argued that similar words appear within similar contexts. The famous quote by Firth [25] captures the idea as; "You shall know a word by the company it keeps!". For instance, the semantic similarity or relatedness between the terms *jacket* and *coat* can be theoretically shown since they will be accompanied by similar verbs, such as *wear*, *dry clean* or *hang*, and similar adjectives such as *warm* or *leather*. We should note that similarity and relatedness differ from each other such that *car* and *motorcycle* are similar to each other while *car* and *road* are related.

Early attempts to show words on a semantic space is studied in Osgood *et al.* [26]. Authors suggested representing concepts using orthogonal scales and relying on human judgement to score meanings on the axes. An example human annotated concept from their study is given in Figure 2.1.

However, for a researcher to pick appropriate scales or have meaning extracted by hand would be infeasible for natural language processing tasks [27].

Even though Harris argued that "language is not merely a bag of words" [24], using just the word counts of a collection to capture the semantic information without regarding the order of the words will be used in the literature and be known as the *bag-of-words* model.

Vector Space Model

The history of word embeddings is tightly coupled with *vector space models* that initially appeared in the field of information retrieval. The intent was to extract vectors that represented *documents*. First vector space model developed by Salton *et al.* [28]

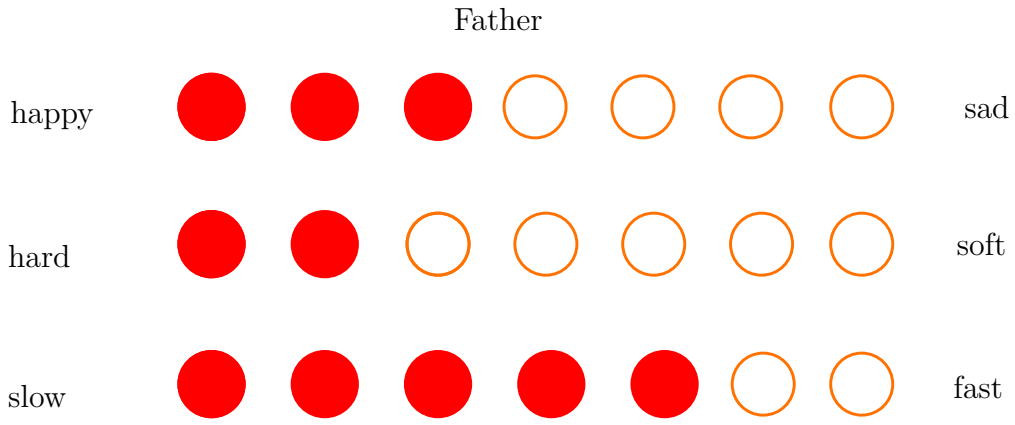

FIGURE 2.1: Sense representation using human judgement scores for the concept "Father" [26]

was presented in "A Vector Space Model for Automatic Indexing". It was the first application of bag-of-words hypothesis on a corpus to extract semantic information [29]. Salton *et al.* presented the novel idea of a *document space*. The document space is built using a term document matrix. The rows of the matrix represent individual documents using the whole vocabulary as their dimension.

In this space, a document D_i is represented using t distinct terms;

$$D_i = (d_{i1}, d_{i2}, \ldots, d_{it})$$

The elements d_{ix} can be raw term counts. They can be weighted using inverse document frequency measure introduced by Jones [30]. Since this weighting scheme uses term frequency and the inverse document frequency, if is shortened as *tf-idf*. *tf-idf* is the multiplication of two metrics;

tf the number of times a term k occurs in a document

idf the inverse of the number of documents that contain k.

The weighting scheme was selected to "assign the largest weight to those terms which arise with high frequency in individual documents, but are at the same time relatively

rare in the collection as a whole" [28].

The vector space model allowed Salton *et al.* to handle the similarity between documents as the angle between two vectors. Cosine similarity measure is often used since inner product of two normalized vectors is equivalent to the angle between them. Salton *et al.* have shown that there is merit to handling documents as real valued vectors.

2.1.2. Latent Semantic Analysis

Deerwester *et al.* [31] introduced latent semantic analysis in order to address a crucial problem with the vector space model. They have identified that the term document matrix approach cannot handle synonyms and homonyms due to the fact that vector space model requires the words to match exactly between the two documents. Synonymity is an issue because the query can have terms that have the same meaning as the target word, without getting matched. On the other hand, homonyms can match with an unrelated word. In order to answer these issues, their model seeks the higher order latent semantic structure in order to learn the *similarity between words*.

Latent semantic analysis starts with a word co-occurrence matrix X. An element $x_{i,j}$ of X is the number of times term i co-occurs with term j in a predefined context. Initially, whole documents were used for the context. Like term document matrices, the terms of the co-occurrence matrix is weighted by some weighting scheme. While original study by Deerwester *et al.* used raw term frequencies, *tf-idf* is a possibility while Levy *et al.* [32] reports pointwise mutual information (PMI) [33] as a popular choice. A term document matrix X is then factorized into three matrices using singular value decomposition [34];

$$X = T_0 S_0 D_0'$$

Where the columns of T_0 and D_0' are orthogonal to each other and S_0 is the diagonal matrix of singular values. The singular values of S_0 can be ordered by size to keep only the k largest elements, setting others to zero [31]. The resulting matrix is shown as S. The similarity tasks that are solved with S can employ the representative k dimensional vectors such that $k << |V|$. Latent semantic analysis was used to the solve document similarity task while *word similarity* was mentioned briefly. Landauer & Dumais [35] later studied word similarity in full using latent semantic analysis, reducing the dimensions of the terms instead of documents.

2.1.3. Building Upon Distributional Hypothesis

While Deerwester *et al.* studied relatedness between words using vectors, their approach used the whole document for the co-occurrence information while the focus was clearly still on the document similarity. Schütze [36] proposed "to represent the semantics of words and contexts in a text as vectors" and built upon word co-occurrence. They theorized a context window of 1000 *characters* in order to consider words that are close to the target word instead of the whole document in co-occurrence calculations. The choice of a character window is justified with the claim that lower number of representative long words are more discriminative than numerous short words. Schütze claimed that the computation power available was not suitable yet to fully tackle the task.

Lund & Burgess [27] took the challenge and experimented with 160 million *words* taken from the Usenet, a precursor to the internet. They used a context window of 10 words and provided a method to obtain feature vectors to represent the meaning of words. However, intricate tuning of word co-occurrence generated associatively similar vectors instead of semantically similar ones.

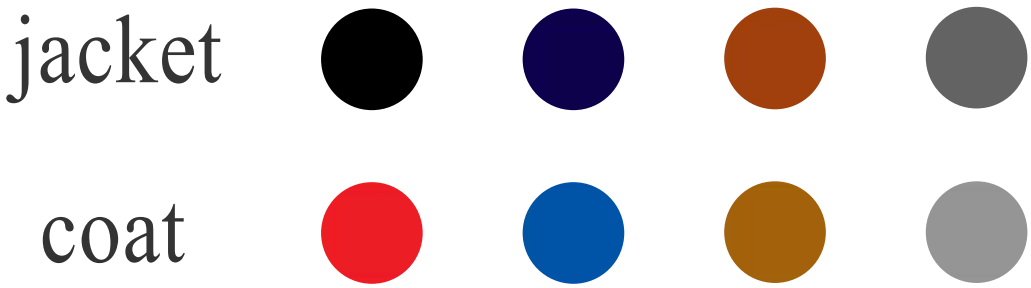

FIGURE 2.2: How semantic similarity can be shown using similarity of two vectors, the colour gradient represents similar values for elements of the vector

2.1.4. Distributed Vector Representations

bengio_neural_2003 proposed learning word representations using a feedforward neural network. Their model learns feature vectors for words using a predictive approach instead of counting based approaches we have presented until now. Although neural networks have been proposed to learn a language model by Xu & Rudnicky [37], the main contribution of **bengio_neural_2003** is to use an *embedding layer*, in order to attack *curse of dimensionality*. Their approach was also motivated exponentially increasing size of the vocabularies caused by *n-grams*. n-grams are representations that are built using multiple tokenized words. For instance, as well as showing *new* and *york* in the vocabulary, *New York* is handled as a single unit. For a corpus with vocabulary V, there are $|V|$ dimensions for the language model to learn and taking *n-gram* representations into consideration, the problem grows exponentially. Using m dimensions in the embedding layer allowed **bengio_neural_2003** to represent words using manageable, more representative dimensions and the problem scaled linearly as a result.

The setup for the neural network starts with the one hot encoded vector representation of the context for a word w, essentially a one dimensional vector where context words of w are set while the rest of the vocabulary are zero. This context window is similar to those used in statistical models that predicts the word w_t using the words that lead up to w_t. In other words, context window of w_t is T words on the left of w_t. In the following equation, we present the problem algebraically with an arbitrary probability function P.

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t|w_1^{t-1})$$

In short, the input layer is projected into an embedding layer, later to a softmax layer to get a probability distribution in order to minimize the following softmax cost function.

$$\hat{P}(w_t|w_{t-1}, \ldots, w_{t-n+1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{2.1}$$

However, this formulation is too computationally expensive since all vocabulary needs to be considered for the sum in the denominator. The curse of dimensionality problem is shifted to the final layer of the neural network. It will be solved later using hierarchical

softmax [22]. Authors reported training times around 3 weeks using 3 to 5 context window sizes and vocabulary sizes around 17000.

Collobert & Weston [19] suggested a deep neural network model in order to learn feature vectors for various natural language processing tasks. Their proposed approach for language model is important for our case since it explicitly learned distributed word representations or simply *word embeddings*. They have introduced two key ideas;

- Instead of using a context window that used words left of the target word to estimate the probability of the target word, they have placed the context window *on* the target window, using n words for left and right of the target word.
- They introduced negative examples, where they randomly changed the middle word with a random one. As well as keeping their model from overfitting, this allowed them to use the ranking cost;

$$\sum_{s \in S} \sum_{w \in D} \max\big(0, 1 - f(s) + f(s^w)\big)$$

2.1.5. Pre-trained Embeddings

P. Turian *et al.* [38] evaluated the performance of different word representations as word features that researchers can include into an existing task. Their contribution is elegantly summarized in their work as;

> Word features can be learned in advance in an unsupervised, task-inspecific, and model-agnostic manner. These word features, once learned, are easily disseminated with other researchers, and easily integrated into existing supervised NLP systems.
> [...]
> With this contribution, word embeddings can now be used off-the-shelf as word features, with no tuning.

2.1.6. Popularization of Word Embeddings

word2vec package [22, 39, 40] popularized word embeddings. There are two aspects of the work done by Mikolov *et al.* that contributed to the fact;

- Their model captures the semantic and syntactic attributes of words and phrases on a large scale with good accuracy, trained on billions of words using a shallow neural network, keeping the computational cost down.
- They published their code as well as their pre-trained embeddings as an open source project[1].

The second point is self explanatory but in order to argue about the first one, we should report the algorithms behind word2vec.

The *skip-gram model* introduced by Mikolov *et al.* [39] differs from the previous methods by predicting the surrounding words given the target word (Figure 2.3).

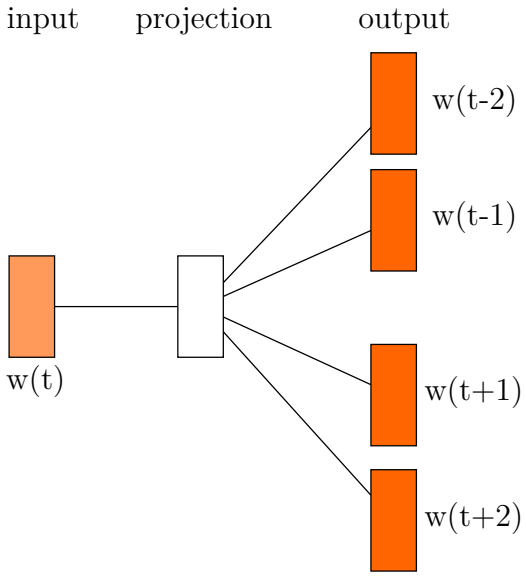

FIGURE 2.3: Skipgram architecture by Mikolov *et al.* [22]

In "Distributed Representations of Words and Phrases and Their Compositionality", it is defined as follows;

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{2.2}$$

The $w_1, w_2, \ldots, w_T$ are the context of the word w_t. We should note that, Levy *et al.* [32] has identified that this window size is *dynamic* in the open source implementation of word2vec, where the actual window size is sampled between 1 and T.

[1] https://code.google.com/archive/p/word2vec

Following word2vec, Pennington *et al.* [23] introduced GloVe embeddings that were built on word co-occurrence probabilities.

Levy *et al.* [32] compared the performance of count based and prediction based word representation models. Representation algorithms they considered are;

- Positive pointwise mutual information (PPMI) [33, 41]
- Singular Value Decomposition on PPMI Matrix (Latent Semantic Analysis) [31]
- Skip-Gram with Negative Sampling [22]
- Global Vectors for Word Representation [23]

They found out that choice of a particular algorithm played an insignificant role compared to choosing the right approach during training, mainly picking correct *hyperparameters*. Besides, the amount of data the models are trained on were more important than the model itself. They used this finding to counter the results reported by Baroni *et al.* [42]. Baroni *et al.* claimed that predictive models outperformed count based models. On the other hand, Levy *et al.* noted that Baroni *et al.* used count based models without hyperparameter tuning, denying them from "tricks" developed in the word representation literature. Finally, Levy & Goldberg [43] emprically proved that word2vec's skip gram with negative sampling approach is equivalent to factorizing a word-context matrix weighted using positive pointwise mutual information. With the amount of overlap and marginal performance gains in between the algorithms, the choice of a particular model seems not as important.

2.1.7. fastText

Armed with the fact that a good word representation model should tune their hyperparameters and should be trained on a large dataset, we set our sights on *fastText*. On their website, authors define fastText as a "Library for efficient text classification and representation learning". The ideas behind it are presented in Mikolov *et al.* [44]. Overall, it builds upon word2vec [22] by adding position dependent features presented in Mnih & Kavukcuoglu [45] and character n-grams suggested on **bojanowski_enriching_2016**.

Let us turn our focus towards **bojanowski_enriching_2016** [**bojanowski_enriching_2016**]. Instead of using a context window to learn the representation of the target word or predicting surrounding words given the centre word like in the the skip-gram model,

bojanowski_enriching_2016 learn representations for character n-grams. We have mentioned that n-grams consider sequences like *New York* as a single token. Character n-grams start with parsing the corpus into c character long sequences. We will use 3 as the c value. The boundaries of the words are marked with "<>" characters for later and the whole corpus is transformed into tokens where the phrase "lecture slide" is now "<lecture> <slide>". Then, sequences of 3 characters are extracted such that "<slide>" is broken down into "<sl, sli, lid, ide, de>". Note that "lid" character n-gram is different from word "<lid>". These character n-grams are called subword vectors, they are trained using the skip-gram architecture.

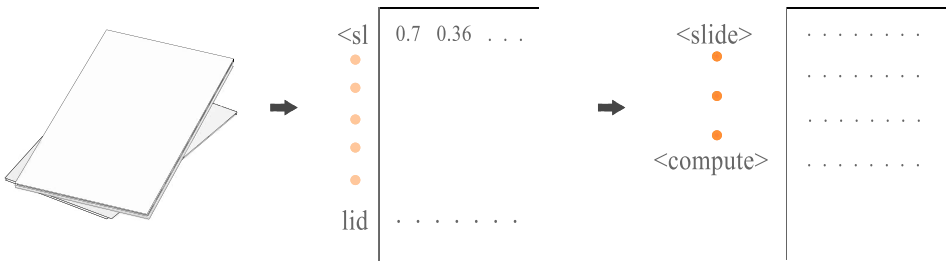

FIGURE 2.4: Overview of character n-gram model

With the subword vectors z_g for every n-gram g at hand, authors take a dictionary of n-grams of size G and for a given word w, they denote $G_w \subset 1, \ldots, G$ as the n-grams of w. So the scoring function for the word in accordance to word's context n-grams is [**bojanowski_enriching_2016**];

$$s(w, c) = \sum_{g \in G_w} z_g^T v_c$$

Grave *et al.* [46] trained fastText model on different languages using Wikipedia and Common Crawl data. Wikipedia is a curated encyclopaedia that can be used as a multilingual corpora with 28 languages that have over *100 million* tokens and 82 languages

with 10 million tokens [46]. Considering the original word2vec was trained on *100 billion* tokens, Grave *et al.* also used Common Crawl, a non profit project that collocates web pages and publishes the data publicly. This data can be inadvertently noisy which Grave *et al.* addressed using linewise language identification and by removing duplicate lines that often appear as leftover boilerplate on many sites. While Wikipedia provided them with a curated signal, Common Crawl data helped with capturing as many contexts as possible to train their distributed model. They are currently hosting the pre-trained word embeddings on their website [2].

2.1.8. ConceptNet Numberbatch

As an alternative to purely distributional models, Speer *et al.* [21] suggested *numberbatch* embeddings built in conjugation with ConceptNet.[3]. ConceptNet is a knowledge graph. Like the WordNet, it presents semantic relationships between concepts but the defined relationships are more fine grained. The relationships are collocated from various resources like English Princeton WordNet, Open Mind Common Sense [47] and DBPedia [48]. ConceptNet stylizes its relations with the `/r/Relationship` syntax. For instance, being distinct members of a set relation is defined as `/r/DistinctFrom` which includes concepts like August and September.[4]. Compared to WordNet, more subjective, human centric relations are defined such as `/r/MotivatedByGoal`, relationship between *compete* and *win* or `/r/ObstructedBy`, the relationship between *sleep* and *noise*. Total number of relations that are available between two concepts is 36. Moreover, these relationships are *weighted* depending on how present they are. For example, the concept *run* is related to *fast* with a weight of `9.42` but the similarity between *race* and *run* is weighted at `2.54`. Finally, the ConcepNet is multilingual, encompassing 304 languages in total but only 10 languages have full support and advised to use for downstream applications. For our case, only Italian is fully supported by numberbatch while the rest of the languages fall into the 77 languages which reported as having moderate support and may be used in downstream tasks with loss in performance. Details of the knowledge graph and the procedure is explained by Speer *et al.* [21] in the paper "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge".

Speer *et al.* also present their resource in word embedding matrix form for ease of use in the current plug-and-play environment. In order to get word embeddings from the

[2] https://fasttext.cc
[3] http://conceptnet.io
[4] https://github.com/commonsense/conceptnet5/wiki/Relations

knowledge graph, first they prepare a symmetric term-term matrix X where an element $w_{i,j}$ is the sum of all edge weights between terms i and j. This is similar to a word co-occurrence matrix but instead of constructing the matrix using unstructured text, this approach builds upon semantic connections between senses. Speer *et al.* reports that their approach learns *relatedness* more so than *similarity* which was reported by the word co-occurrence approaches before [27].

With the term-term matrix at hand, Speer *et al.* weigh the terms using positive point-wise mutual information (PPMI) as suggested by Levy *et al.* [32] and reduce it to 300 dimensions, as is standard set by word2vec, using truncated SVD. Resulting matrix is similar to approaches set by Deerwester *et al.* [31] or Pennington *et al.* [23] but Speer *et al.* enrich it further using *retrofitting* as proposed by Faruqui & Dyer [49]. Pre-trained word2vec and GloVe embeddings are incorporated into the reduced term-term matrix to finally obtain embeddings that include both distributional and semantic relatedness signal. Authors call their finalized model *ConceptNet numberbatch*. Speer *et al.* reports state of the art results compared to word2vec embeddings on word relatedness and models built using their embeddings get scores equivalent to humans on SAT-style analogy tasks.

2.2. Approaches in Wordnet Generation

2.2.1. History of Wordnet Generation

We have mentioned the lexical database WordNet created by Princeton University. To reiterate, WordNet is a lexical database with human annotated collection of senses and relationships among them. The relationships are hierarchical so they can be followed along to reach new nodes due to the transitive property (shown in Figure 1.2). The format itself has become the standard for databases that present meanings and concepts [50].

Glosses or the definitions that go along with synsets were not initially part of the WordNet design. Authors believed that "definition by synonymity" would be enough. In other words, definition of a synset can be derived from the lemmas that make up the synset. As the number of items in the WordNet grew, only then short glosses, later followed by longer definitions got included in WordNet [6].

Synset	Gloss
{glossary, gloss}	an alphabetical list of technical terms in some specialized field of knowledge; usually published as an appendix to a text on that field
{dog, domestic dog, Canis familiaris}	(a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds)
{university}	(the body of faculty and students at a university)
{depository financial institution, bank, banking concern, banking company}	(a financial institution that accepts deposits and channels the money into lending activities)

TABLE 2.1: Example synsets and their respective glosses from WordNet

WordNet has been used in various natural language processing applications over the years such as text summarization [51] or word sense disambiguation [52]. Since the original WordNet was prepared for English over many years of work, efforts for creating an equivalent resource for other languages has been initiated. Arguably, EuroWordNet set the standard for creating wordnets for languages other than English [**vossen_introduction_1998**, 53].

EuroWordNet[5] project was initiated to introduce the benefits of English Princeton WordNet for other languages. Additionally, an interlinked semantic network can be a research topic on lexicalization patterns of languages, finding conceptual clusters of vocabularies or cross lingual text retrieval [**vossen_introduction_1998**, 54]. EuroWordNet project included 7 wordnets for languages other than English and an adapted English wordnet. Due to the effort needed to create a wordnet from scratch, **vossen_introduction_1998** averted the EuroWordNet project from creating full scale semantic lexicons and prioritized the connectivity between wordnets. All in all, **vossen_introduction_1998** defines the aims as;

1. to create a multilingual database;
2. to maintain language-specific relations in the wordnets;
3. to achieve maximal compatibility across the different resources;

[5] http://projects.illc.uva.nl/EuroWordNet

4. to build the wordnets relatively independently (re)-using existing resources;

One challenge in achieving compatibility is the shortcomings of using the original WordNet as the anchor. On one hand, since the WordNet is the first and the most comprehensive, it is a natural hub for new wordnets. On the other hand, a sense in one language might not have a direct equivalent in an other. Cultural differences or linguistic differences between languages contribute to this fact [55] which is called a lexical gap or untranslatability. EuroWordNet addresses lexical gaps using Inter-Lingual-Index (ILI). ILI is a higher order list of meanings just for wordnet synsets to align themselves to, elevating the burden for alignment from English Princeton WordNet. The introduction of ILI allowed language specific structures to exist in wordnets while keeping the connections among themselves.

Two approaches for wordnet generation were defined by **vossen_introduction_1998**;

Merge Approach where a wordnet structure is formed in the target language with synset selection and relation mapping. Then the connections between the new wordnet and English Princeton WordNet can be established.

Expand Approach where English Princeton WordNet is (machine) translated to target language [56], preserving connection information with a trade-off where the target language wordnet will be biased towards the relationships of the English Princeton WordNet and may not include target language specific lexical connections.

In order to maintain as much language specific lexical connections as possible while having a starting point for evaluation of target wordnets, EuroWordNet project offered "Base Concepts". This idea evolved into 1000 and later to 5000 core synsets that are compiled from most frequent, connected and representetive synsets to be used for evaluating wordnet generation [57].

2.2.2. Examples of Wordnet Generation

Following the EuroWordNet project, several studies were published on wordnet generation with proposals heavily leaning towards the expand approach. **diab_feasibility_2004** explored whether an Arabic wordnet is attainable or not using a parallel corpora.

Arabic differs from the languages explored in EurowordNet due to its unusual morphological nature. Using their proposed method, they observed 52.3% of words they processed are sufficient for a future Arabic wordnet. They have also reiterated that semantic relationships in a language are transferable to a target language's wordnet.

Further approaches using parallel corpora to align the target language with English Princeton WordNet were used by **sagot_building_2008** in order to create a production ready French wordnet and by Fiser [58] to create a Slovene wordnet.

Approaches that used machine translation to get potential synsets for the target language was explored by Lam *et al.* [59]. They proposed two approaches for this task. First approach uses a single bilingual dictionary to translate English Princeton WordNet lemmas to target language to form synsets. Second approach translates existing wordnet synsets to English and then translates them to target language.

Following the publication of word2vec, approaches that use word embeddings gained traction and are interest to us. **sand_wordnet_2017** used word embeddings to extend Norwegian wordnet by adding new relationships on existing synsets or by introducing new synsets all together. First, cosine similarity measure is used to discover nearest neighbours to a potential synset. Then a threshold value is used to cut off any new synsets below a certain similarity value. They evaluated their approach against accuracy, the percentage of relations that were correct and attachment which is similar to a recall score. It is the percentage of considered synsets that were above the threshold value. Overall, **sand_wordnet_2017** reported accuracy scores within the 55.80 to 64.47 percent range with respect to different frequency thresholds and an attachment score that fluctuate between 96.20 and 98.36 percent.

Arguably most relevant to our study, Khodak *et al.* [60] proposed an unsupervised method for automated construction of wordnets. In their paper "Automated WordNet Construction Using Word Embeddings" they present 3 approaches and compare their precision, recall and coverage.

First off, they picked 200 French and Russian adjectives, nouns and verbs, totalling 2 corpora with 600 items in each. These words are selected based on if they have a translation in the core sense list provided by English Princeton WordNet. Their approach starts with the processing of a word w. Initially, the presented method collects the possible translations of w using a bilingual dictionary and uses them as lemmas to query English Princeton WordNet. As we have mentioned, lemmas that

query the WordNet retrieve synsets in the form `<lemma.pos.offset>`. Furthermore, every synset includes possible lemmas that can represent the sense of the synset (Refer to Table 2.1). These lemmas form the set S. By translating the English Princeton WordNet lemmas to target language, they obtain the set T_S Elements of T_S and w are both in target language so a monolingual word embeddings can be used; As a baseline, they calculate the average cosine similarity between word embeddings of every element of S and w.

$$\frac{1}{S}\sum_{w' \in S} v_w \cdot v_{w'}$$

In order to improve the discriminative power of their target vectors, they use synsets relations, definitions and example sentences. These are used to get a more representative vector v_L. Given resources can be thought as sentences and sentence embeddings are calculated by getting a weighted average of word vectors of words that make up the sentence. The weighting is called smooth inverse frequency which was suggested by **arora_simple_2016**;

$$v_L = \sum_{w' \in L} \frac{a}{a + Pw'} v_{w'}$$

We will talk about sentence embeddings in detail for our approach in Section 3.1.

2.3. Other Related Work

Lesk [61] presented the *Lesk Algorithm* which is one of the earliest answers to word sense disambiguation problem. This classic algorithm identifies the correct sense for a couple of words among their synonyms using the word that it co-occurs with. The classic example uses the *pine cone* for demonstration. First, the dictionary definitions of the two words in the sentence are extracted. For *pine*, the definitions are "kind of evergreen tree with needle-shaped leaves …" and "waste away through sorrow or illness …". For *cone*, the definitions are presented as "solid body which narrows to a point …", "something of this shape whiter solid or hollow …" and "fruit of certain evergreen trees" [61]. The algorithm then assigns the sense pairs that have the highest number of word overlaps between them. For the *pine cone* case, the algorithm finds the overlap on

evergreen and *tree* to assign the matching senses; "kind of evergreen tree with needle-shaped leaves …" and "fruit of certain evergreen trees". This approach relied on the definitions given by traditional dictionaries.

Banerjee & Pedersen [52] extended the *Lesk Algorithm* by levering the WordNet synsets. Instead of comparing the definitions of the pair of words, their approach uses the definitions of the co-occurring words as well as the definitions of synsets that are connected to the word pair.

Gordeev *et al.* [62] presented various approaches for matching cross lingual product classifications or product taxonomies using bilingual embeddings. Their work is highly related since they have also studied the viability of showing senses using word embeddings on a latent space which were Russian and English product descriptions. However, they relied on other information sources such as the category of the products which is absent in our study. Our study and their approaches are similar in an additional way; they reported some products did not have direct equivalents on the target language's product space, which highly resembles the lexical gap we have talked about in Section 2.2. They also used out of domain pre-trained word embeddings due to small size of their corpora.

Three algorithms that were presented which are relevant to our study;

1. doc2vec [63] on untranslated text.
2. doc2vec on translated text.
3. averaging the category description vector.

Gordeev *et al.* reported 19.75% accuracy on the first approach and 44% on the second approach. Like our case, their best results came from using out of domain word embedding vectors to encode the text at hand, which is presented in the third approach with an accuracy of 55%.

3. Unsupervised Matching

3.1. Linear Assignment Using Sentence Embeddings

Word embeddings represent single tokens or n-grams such as *San Francisco* depending on the implementation. Yet, word2vec suggested and proved that word embeddings are compositional [22]. Research then moved on to study the representation of longer pieces of text like documents, paragraphs and sentences. Implementations like Le & Mikolov [63] extended the skip-gram idea from Mikolov *et al.* [22] to learn the feature vectors for *paragraphs* and used them to predict surrounding paragraphs in the text. Approaches like Kiros *et al.* [64] trained an encoder that constructed surrounding sentences to learn sentence representations. However, corpora in question for the given studies are continuous. When the document collection we would like to show on a latent space is not part of a longer text but occur as *discrete* pieces, that assumption does not hold. Regarding dictionary definitions, we cannot rely on continuous models. Our dictionary definitions are written with of 10 to 11 words each with no relation from one distinct dictionary definition to another. Refer to Table 3.1 for examples of a typical collection of definitions.

With such short pieces of text, considering the scope of a possible model could have, instead of paragraph embeddings we can talk about *sentence embeddings*. A sentence embedding model should ideally capture the collective meaning of the short text where every word is potentially informative and discriminative.

Wieting *et al.* [65] studied sentence embeddings and reported that averaging word embeddings that make up a sentence to get sentence embeddings is a valid and surprisingly effective approach. **arora_simple_2016** built upon the simple model and has shown that weighed average of word vectors perform so well, their publication is called; **arora_simple_2016**. In the suggested approach, word embeddings that make up a sentence is weighed with a scale called *smooth inverse frequency* and averaged across the sentence. Smooth inverse frequency is suggested as

$$\text{SIF (w)} = \frac{a}{a + p(w)}$$

where a is a hyperparameter and $p(w)$ is the estimated word probability [**arora_simple_2016**].

Using smooth inverse frequency weighting, word embeddings $v_w \in R^d$ where word w is in a vocabulary V can be averaged over a sentence S such that $S \subset V$ to get sentence embedding v_S in the same dimensionality R^d.

$$v_S = \frac{1}{|S|} \sum_{w \in S} \text{SIF (w)} v_w \tag{3.1}$$

The authors point out that the metric is similar to *tf-idf* weighting scheme if "one treats a 'sentence' as a 'document' and make the reasonable assumption that the sentence doesn't typically contain repeated words" [**arora_simple_2016**]. These assumptions hold for us so we scaled our word embeddings using *tf-idf* weights to get *sentence embeddings*.

$$v_S = \frac{1}{|S|} \sum_{w \in S} \text{tf-idf}_{w,S} v_w \tag{3.2}$$

Parallel to **arora_simple_2016**, Zhao *et al.* [66] used two approaches for sentence embeddings in order to solve SemEval-2015 Task 2: Semantic Textual Similarity [1]. First, for a sentence $S = (w_1, w_2, \ldots, w_s)$ where the length of the presumably small sentence is $|S| = s$ and the word embedding of a word w is v_w;

- They summed up the word embeddings of the sentence $\sum_{t \in S} v_t$
- Used information content [67] to weigh each word's LSA vector $\sum_{w \in S} I(w) v_w$

Both approaches results in a vector that is in the same dimensions R^d as the original word representations.

Edilson A. Corrêa *et al.* [68] expanded upon this simple yet effective idea to tackle the SemEval-2017 Task 4[2], Sentiment Analysis in Twitter. We have mentioned the discrete short pieces of text that are present in our definition corpora. *Twitter* exhibits a very similar case to our study.[3] *Tweets* are short pieces of text due to the 280 character constraint imposed by the platform. In order to acquire embeddings that represented *tweets*, they weighed the word embeddings that made up a tweet; $\text{tweet}_i = (w_{i1}, w_{i2}, \ldots, w_{im})$ with the *tf-idf* weights. For the *tf-idf* calculation, they cast individual weights as documents so that term frequency become the term count in

[1] http://alt.qcri.org/semeval2015/task2
[2] http://alt.qcri.org/semeval2017/task4
[3] https://twitter.com

turn red, as if in embarrassment or shame
a feeling of extreme joy
a person who charms others (usually by personal attractiveness)
so as to appear worn and threadbare or dilapidated
a large indefinite number
distributed in portions (often equal) on the basis of a plan or purpose
a lengthy rebuke

TABLE 3.1: Some definitions from English Princeton WordNet

a single tweet while document frequency become the number of tweets the term w_t occurs.

For the *tf-idf* calculations, we followed a similar approach. The term frequency is the raw count of a term in a dictionary definition. While the document frequency is the number of dictionary definitions where w occurs.

Then, with a term embedding matrix at hand, we have calculated definition embeddings using;

$$S_{\text{emb}}(S) = \sum_{t \in S} tf_{t,S}\text{-}idf_t \cdot Emb_w(t) \tag{3.3}$$

Every word that makes up a definition is scaled by its vector in $\mathbb{R}^n$, then concatenated to form sentence embeddings on $\mathbb{R}^n$.

As we have N definitions in source wordnet (English Princeton WordNet) and target wordnet we have accepted as golden or perfectly aligned, we now hypothesize that there exists a one-to-one mapping between two sets. In order to discover this mapping, we can get leverage the sentence embeddings we just got and cast the cosine similarity between two sentence embeddings across languages as the weight between them. Given N real valued vectors from source and target wordnet this problem naively iterates over $N!$ matchings to find the case where the sum of the similarity is maximum. Our problem is an instance of an linear assignment problem.

3.2. Linear Assignment Algorithm

Linear assignment problem is an optimization problem where a cost matrix is solved. A cost matrix is formulated such that each row corresponds to a task and each column

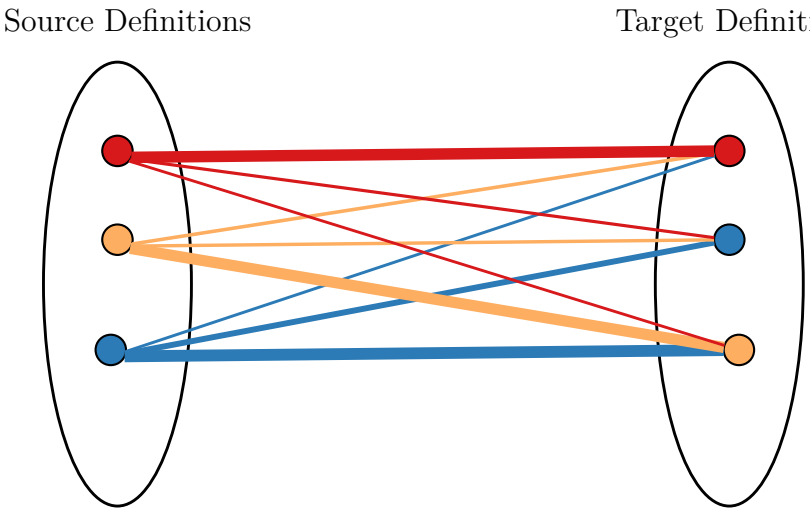

FIGURE 3.1: Matching sentence embeddings can be shown as a variant of finding the maximum flow in a bipartite graph. In the figure above, the connections denote the similarity between the sentences and the width of the stroke represents the magnitude of that similarity between two definition nodes. Any two pairs of definitions have some similarity defined between them yet the matched definition are picked to ensure the overall flow is maximum. In the figure above, matched nodes have the same colour. Note the blue node, it is not assigned to the most similar sentence in order increase the overall similarity between the two disjoint sets.

corresponds to a worker for the task. The aim is to find an assignment such that each task is solved by one worker and the total cost is minimized.

In our case, this formulation corresponds to two sets of wordnet definitions. The definitions are disjoint and independent with respect to their parent wordnet. The weights among the sets are the similarity between individual dictionary definitions. Refer to Figure 3.1 for a representation of this notation.

We have calculated sentence embeddings such that each node of the graph is an d dimensional sentence vector $v_S \in R^d$. The weights that connect one definition to another in this bipartite graph is the cosine similarity between two sentence's sentence vectors. Cosine similarity is constrained in $[0, 1]$ where 1 is perfect similarity and 0 denotes two orthogonal vectors, or no similarity. This is crucial for our task such that we set out to maximize the total weight in this matching.

One of the most famous solvers for the linear assignment problem is the Hungarian method [69]. Given a qualification matrix, the Hungarian method uses a heuristic to solve the assignment in $O(n^3)$ time. We use Jonker & Volgenant [70] solver for the problem. Jonker & Volgenant improve upon the Hungarian Algorithm using initialization heuristics and shortest path algorithm of Dijkstra [71]. The time complexity is still $O(n^3)$ but in real life problems, Jonker & Volgenant is faster [72].

3.2.1. Creating The Cost Matrix

The creation of the cost matrix starts with the bilingual word embedding matrix W. W is $m \times d$ where d is the dimensionality of the word embeddings and m is the size of the joint vocabulary across two wordnets $|V|$. The definitions are parsed into term document matrices such that we obtain two matrices T_s and T_t for source and target dictionaries. We weigh T_s and T_t using *tf-idf* as we have mentioned before, the term count is the number of times a term occurs in a definition and document frequency is the number of definitions that include the term. The resulting matrices are T'_s and T'_t. In order to get sentence embedding matrices, first we normalize the term document matrices and run matrix multiplication such that;

$$S_x = T'_x \times W \tag{3.4}$$

using this equation, we obtain S_s and S_t for source and target sentence embeddings. By multiplying of S_s and the transpose of the S_t, S_t^T, we obtain a cost matrix C that

is immediately compatible with the linear assignment solver. An element $c_{i,j}$ of C is the cosine similarity between source definition i and target definition j.

3.2.2. Evaluation

The matching approach is the application of our hypothesis that there exists a one-to-one matching between two sets of dictionary definitions. As a natural extension of this, matching definitions across wordnets results in a single *answer* for each *query*. Hence we will report precision at one scores for matching approaches. Precision at one is the fraction of correct matches throughout the experiment set. In Chapter 6, we will present it as a percentage rather than fraction for legibility.

The matching approaches require two sets of dictionary definitions to have same the number of definitions in both sets, or the sets should have the same *cardinality*. For an experiment run with cardinality C, say the number of correctly matched definitions are q. Then we calculate our score using;

$$P = \frac{q}{C}\% \tag{3.5}$$

4. Dictionary Alignment as Pseudo-Document Retrieval

Document retrieval is the prototypical information retrieval task. Bush [73] theorized the possibilities of the automatic information retrieval by machines in his essay titled "As We May Think". On his "Modern Information Retrieval", Singhal [74] gives due credit to Luhn [75] for the initial suggestion of using word overlap in document retrieval.

Modern information retrieval techniques are out of the scope of this thesis, our short definitions are arguably not even documents. Yet, we can still benefit from the tried and tested methods of the early information retrieval. Considering the small collection of documents at hand, first we will investigate if we can handle the task using approaches that were available to the researchers when the size of corpora that were available to them were small as well [74].

4.1. Google Translate Monolingual Baseline

First of all, by collecting English Princeton WordNet definitions and target wordnet definitions, we created a corpora suitable for the task by translating the target language's definitions to English using Google Translate. We used the Google Cloud API[1] in order to automate the process.

Table 4.1 presents some definitions and their automated translations to English. While the translations can inform the user about the general sense of the word, sentence structure is flawed at times. We can also see word duplication when synonyms collapse into single words. These shortcomings are also reported by Groves & Mundt [76].

With the English Princeton WordNet definitions and 6 target wordnet definitions at hand, we can handle the task as *monolingual document retrieval*. As usual, we present our methods with a single wordnet which is repeated for 6 wordnets trivially. We will use the vector space model that was mentioned in Chapter 2 to have real valued vectors that represent the definitions.

[1] https://cloud.google.com/translate

Original Definition	**Translated Definition**
bila pri starih Grkih in Rimljanih vojna ladja s tremi vrstami vesel	with the ancient Greeks and Romans, a three-wheeled war ship was happy
znanstvena veja matematike, ki se ukvarja s prostorskimi lastnostmi teles in njihovimi medsebojnimi odnosi	the scientific branch of mathematics, which deals with the spatial properties of the bodies and their interrelations
circumstanțe, stări de fapte, lucruri luate în calcul într + o discuție sau dezbatere	circumstances, facts, things taken into account in a discussion or discussion
Fază a lunii în care este complet iluminată	Phase of the month in which it is fully illuminated
Persoană care se ocupă cu pescuitul și uneori cu conservarea peștelui pescuit	A person who deals with fishing and sometimes with the conservation of fish fishing
E bëj diçka që të shkojë e të përputhet me një qëllim të caktuar, i bëj ndryshimet e ndreqjet e nevojshme, që t'u përgjigjet kushteve e rrethanave të caktuara.	I do something to go and match a certain purpose, make the necessary adjustments and adjustments, to respond to certain conditions and circumstances.
pohoj, a paraqit dikujt një gjë për ta parë, për t'u njohur me të a për të gjykuar për të; zbuloj diçka që të shihet; tregoj	I assure you, did someone present a thing to see, to know him or to judge him? I find something to be seen; show

Table 4.1: Example definitions and translations

4.1.1. Term Document Matrix

With the corpora of two wordnets in English, we have can use a single vocabulary V. With the definition collection and the vocabulary at hand, we create a *term-document* matrix. In this instance of the term-document matrix, rows represent the documents and columns denote individual vocabulary entries. Such a matrix C is m by n while m is the number of documents and n is the size of the vocabulary $|V|$. Our definitions are used as short documents so we use the terms *definition* and *document* interchangeably with the added benefit of using d to denote either of them.

For any element of C, $c_{i,j}$ is the number of times j occurs in the document i. This matrix is sparse since documents use a fraction of the available vocabulary V. Our case with definitions is no different with an average of 10.62 words per definition with vocabulary sizes well over 10,000.

4.1.2. Term Weighting

Elements of a term-document matrix is scaled in order to assign weights to elements according to their discriminative power [77]. Stopwords like "the", "and" have virtually no significance and terms that are common in the corpora's domain will have no discriminating power either [77].

What is a good scaling method for term weighing? *tf-idf* is composed of two scores; term frequency *tf* and inverse document frequency *idf*. *idf* was initially suggested by Jones [30] and is highlighted by Manning *et al.* [77]. Term frequency $tf_{w,d}$ is the number of times term w occurs in a document d. Inverse document frequency idf_w is calculated using the number of documents that contain the term w which is also known as the document frequency of w, df_w [77] We have used the smooth variant of idf as follows;

$$idf_w = \log \frac{N+1}{df_w + 1} \tag{4.1}$$

Where N is the number of documents in the corpus. $\log(\cdot)$ is used to dampen the effect of the *idf*. Using *idf* ensures that terms that appear rarely have a bigger influence and coupled with *tf*, the highest weights are given to rare terms when they appear numerous times in a document. On the other hand, terms that appear commonly in documents or even in every document lose their influence [77]. By weighing our term-document

matrix C, for any term $x_{i,j}$

$$x_{i,j} = \text{tf}_{td} \cdot \text{idf}_t \tag{4.2}$$

term $x_{i,j}$ shows the importance of term t with relation to its general significance throughout the corpus. Our choice of *tf-idf* weighting is further motivated by Ruiz-Casado *et al.* [78] in which they claim superior results in discovering correct wordnet definitions using *tf-idf* weights.

4.1.3. Similarity Measure

With our term-document matrix ready, similarity between two documents can be calculated by assigning a scoring scheme that uses document vectors. *Cosine similarity* is a measure that assigns to find the cosine of the angle between two vectors as their similarity score and scales to multidimensional case trivially. Moreover, since the majority of the dimensions between the vectors are zero, cosine similarity ignores these dimensions and implicitly considers the non-zero dimensions.

For the row vector $\vec{d_t}$ and $\vec{d_p}$, cosine similarity between definitions t, p is

$$\cos(\theta) = \text{sim}(d_t, d_p) = \frac{\vec{d_t} \cdot \vec{d_p}}{|\vec{d_t}||\vec{d_p}|} \tag{4.3}$$

4.1.4. Retrieval

With definition vectors and a similarity measure at hand, we split the *tf-idf* weighted term-document matrix C' as English Princeton WordNet definitions and target wordnet definitions. The target wordnet definitions were used as pseudo-queries and English Princeton WordNet definitions were used as the document collection to retrieve definitions from. In Figure 4.1, we show the overview of the process. As we have mentioned in Chapter 3, the golden corpora we use includes the same number of definitions on both collections. This number is shown as k in Figure 4.1.

In order to retrieve and rank definitions given the query definition, cosine similarity between query and each corpus definition is calculated. The similarity scores that range between 0 and 1 are sorted in descending order with the definition index attached. The retrieved documents are ranked according to their similarity score. Since the definitions are aligned across wordnets, a correctly retrieved definition will have the same index

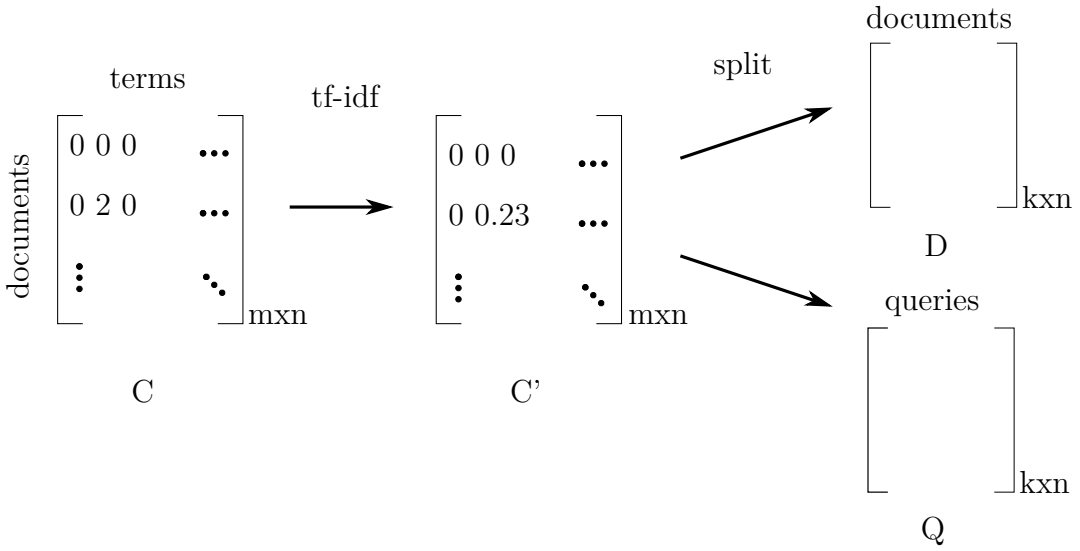

FIGURE 4.1: Term document matrix to queries and documents

as the query definition. By sorting the retrieved indices of the definitions, we have a ranking system.

4.1.5. Evaluation

Information retrieval literature usually uses *precision* and *recall* as the evaluation metric with the assumption that the user wants to receive as many relevant documents as possible [79]. However, we only have one correct definition we are interested in and cannot access relevance any further for any other retrieved definitions anyway. The only information our corpora provides is that *correct* definition to retrieve has the same index as the query. So, we used a tweaked version of *Mean reciprocal rank* (MRR) as introduced by Voorhees [80]. In their work, mean reciprocal rank was used in order to evaluate a question answering track where a singular correct answer was sought among several answers and is rewarded according to the rank of the correct item. In other words, mean reciprocal rank evaluates a system's retrieval score with respect to the rank of a single correct answer, averaged over all the queries. Considering that we have k definitions that we cast as queries, the MRR is calculated as

$$\text{MRR} = \frac{1}{k}\sum_{q=1}^{k}\frac{1}{R(q)} \tag{4.4}$$

Where rank_q is the ranking of the *correct* definition for the query q.

English Princeton WordNet Definition	**Translated Definition**
ancient Greek or Roman galley or warship having three tiers of oars on each side	with the ancient Greeks and Romans, a three-wheeled war ship was happy
the pure mathematics of points and lines and curves and surfaces	the scientific branch of mathematics, which deals with the spatial properties of the bodies and their interrelations
everything stated or assumed in a given discussion	circumstances, facts, things taken into account in a discussion or discussion
the time when the Moon is fully illuminated	Phase of the month in which it is fully illuminated
someone whose occupation is catching fish	A person who deals with fishing and sometimes with the conservation of fish fishing
make fit for, or change to suit a new purpose	I do something to go and match a certain purpose, make the necessary adjustments and adjustments, to respond to certain conditions and circumstances.
admit (to a wrongdoing)	I assure you, did someone present a thing to see, to know him or to judge him? I find something to be seen; show

TABLE 4.2: English Princeton WordNet definitions and the target wordnet definitions we want to match

4.2. Cross Lingual Document Retrieval

In this section, we will talk about the methods that allowed us to use bilingual word embeddings to retrieve definitions while keeping them in their own language.

4.2.1. Word Movers Distance

Following the popularization of the word2vec [22], **kusner_word_2015** introduced *word mover's distance*. In order to explain the algorithm, first we should talk about the motivation behind it.

We have studied bag of words representation for document retrieval tasks, using *tf-idf* weights. **kusner_word_2015** gives a brilliant example where relying on word overlap falls short;

Obama speaks to the media in Illinois	The President greets the press in Chicago

After the stop words are removed, the resulting tokens {Obama, speaks, media, Illinois} and {President, greets, press, Chicago} have no overlap so their vector representations would be orthogonal, showing maximum dissimilarity. Yet they convey the same meaning semantically. By calculating the distances between individual words of documents, it can be proven that the case above is basically the same sentence. Using this case as their motivation, **kusner_word_2015** adapted the optimal transport theorem [**kantorovitch_translocation_1958**], an optimization technique like the one discussed in Chapter 3. This theorem deals with the energy cost of transporting one arbitrary mass to another, often explained with transporting piles of dirt analogy. Their idea starts by casting documents as probability distributions defined over word embeddings of the words they are written with. First, words of a document is thought as the elements of a document's probability distribution. We now know that relying on sameness or word overlap is not feasible so the words are instead represented by their d dimensional word embeddings. The distance between two words can be represented by the Eucledian distance between them over k dimensions, or as the distance metric $c(\cdot)$;

$$c(w_a, w_b) = ||w_a - w_b||_2$$

kusner_word_2015 calls this the travel cost between word w^a and w^b. Now that the distance or the cost between words can be shown, two documents d_i and d_j, respectively written using words such that $d_i = \{w_1^i, w_2^i \dots w_n^i\}$ and $d_j = \{w_1^j, w_2^j \dots w_n^j\}$, a *flow matrix* can be formulated using the distances between every w_x^i and w_y^j pair. This flow matrix is called T and an element $T_{i,j}$ is how much of the distance needs to be travelled when starting from word i to reach word j in order to move d_i to d_j. The T is optimized to find the minimum distance possible between d_i and d_j using linear programming. Mainly, the following constraints are solved in order to find the T with the least overall sum [**kusner_word_2015**];

$$\sum_{i,j}^{n} T_{i,j} c(i,j) \tag{4.5}$$

$$\sum_{j=1}^{n} T_{i,j} = d_i \forall i \in 1, \ldots, n \tag{4.6}$$

$$\sum_{i=1}^{n} T_{i,j} = d'_j \forall j \in 1, \ldots, n \tag{4.7}$$

This optimization is solved by *choosing* which word pairs to use to travel from one document to another and when cast over a whole corpus. Similar documents will have lower minimal costs on their T matrices depending if their words are easily *moveable* across each other.

4.2.2. Cross Lingual Word Mover's Distance

Balikas *et al.* [20] extended the word mover's distance with two additions. First, instead of normalized bag of words representation for the documents, they used term frequency and *tf-idf* to weigh the document representations. Second, they introduced *entropic regularization* which allowed them to use Sinkhorn-Knopp algorithm [**sinkhorn_concerning_1967**] to solve the linear assignment between word embeddings of the source and the target documents.

We have modified their open source project[2] to accompany our task.

4.2.3. Evaluation

In order to evaluate cross lingual pseudo document retrieval approaches, mean reciprocal rank that was presented in Equation 4.4 is reported here as well. Mean reciprocal rank gives an overall insight about the performance of the retrieval system by giving penalty to cases that retrieved the correct definition in a lower rank. However, in the context of dictionary alignment, percentage score of *precision at one* is also reported since in a real life application, retrieving the correct definition on the top spot is more important.

[2] https://github.com/balikasg/WassersteinRetrieval

5. Supervised Alignment

The approaches we have presented so far work fully unsupervised. Given two collections of definitions, we have studied the methods that retrieved the definition(s) that represented the same meaning. In this chapter, given the moderately sized data in our hands we have accepted as "golden", we will investigate the feasibility of training an encoder [81], where the objective is to learn whether the pair of definitions entail the same sense across languages.

5.1. Neural Network Model

Recurrent Neural Network (RNN) architectures improve upon the prototypical neural network model by introducing a *memory* for the connections in the network [82]. By updating the hidden unit over time using the output of the previous time step, the model can *remember* features of the input signal for later inputs. One particular architecture of RNNs propsed by Hochreiter & Schmidhuber [83] is *long short-term memory* (LSTM). LSTM models have been successful on language modelling tasks [81], handwriting recognition [84, 85] and machine translation with a focus on rare words [86]. Highlight of these results are that LSTM has an advantage on tasks that require contextual information to persist over long periods of time [87]. Furthermore, LSTMs do not require fixed input vectors, which is a necessity for us since our definition pairs do not have to be the same size.

5.1.1. Vanishing Gradient Problem

LSTM is born out of the need to address the *vanishing gradient problem* [83, 88]. On the original publication by Hochreiter & Schmidhuber [83], a crucial shortcoming of RNNs have been identified as their slow rate of training which may not converge in the end at all. Independently, Bengio *et al.* [88] suggested that the problem stems from the choice between the conservation of the previous inputs versus resisting against the noise they accumulate. Figure 5.1 illustrates the problem using shades as the influence of input over neural network units. As the input signal traverses the units of an RNN, it either diminishes or blows up [87].

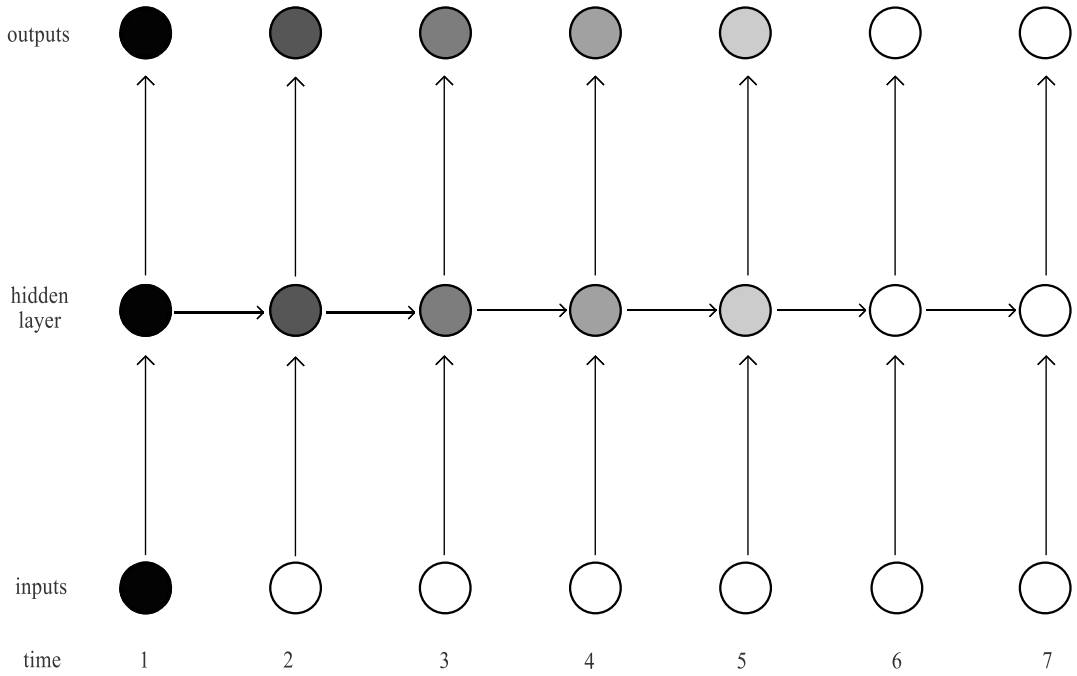

FIGURE 5.1: Graphical representation of vanishing gradient problem where the shades of the nodes represent the influence of the input signal [87]

5.1.2. Long Short-Term Memory

LSTM is the solution highlighted by Graves [87] as a recurrent neural network model that can work over temporally distant input signals while preserving their influence or diminishing their noise. The centrepiece idea is to use a *constant error carousel*, special cells that enforce a constant error flow. This complex unit is named *memory cell* [83]. Using an *input gate*, the cell is updated if the current input is relevant and using an *output gate*, the unit will not update other cells if the current output is not relevant. A simplified overview of the suggested model is presented in Figure 5.2.

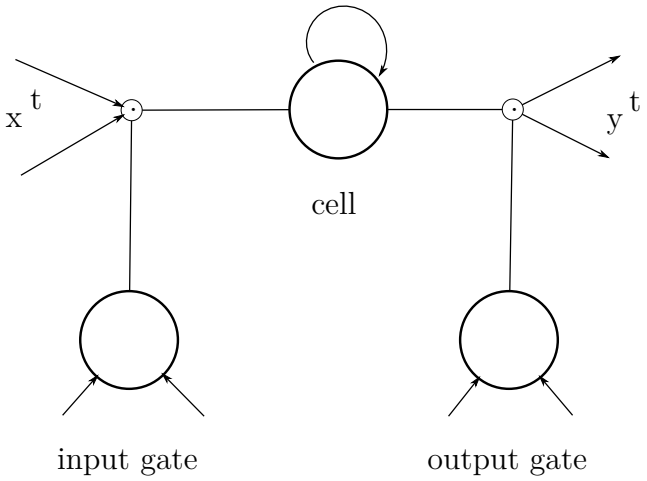

FIGURE 5.2: Simplified long short-term memory cell architecture

Multiple arrows denote input from current time frame and recurrent connections. $\odot$ symbol denotes multiplication. By weighing the input and output gates between 0 and 1, the impact of the current input and output can be adjusted. Overall, input gate controls how much cell will learn and output gate controls how much the cell will propagate. Figure 5.3 adapted from Graves [87] illustrates how the LSTMs operate against vanishing gradient.

Two gates on top of the cell structure model got extended with a third *forget gate* by Gers *et al.* [89]. The aim was to handle input sequences that are not segmented in a predictable manner. The error signals were getting carried too far back in time which deteriorated the performance on tasks with continuous input streams. The proposed

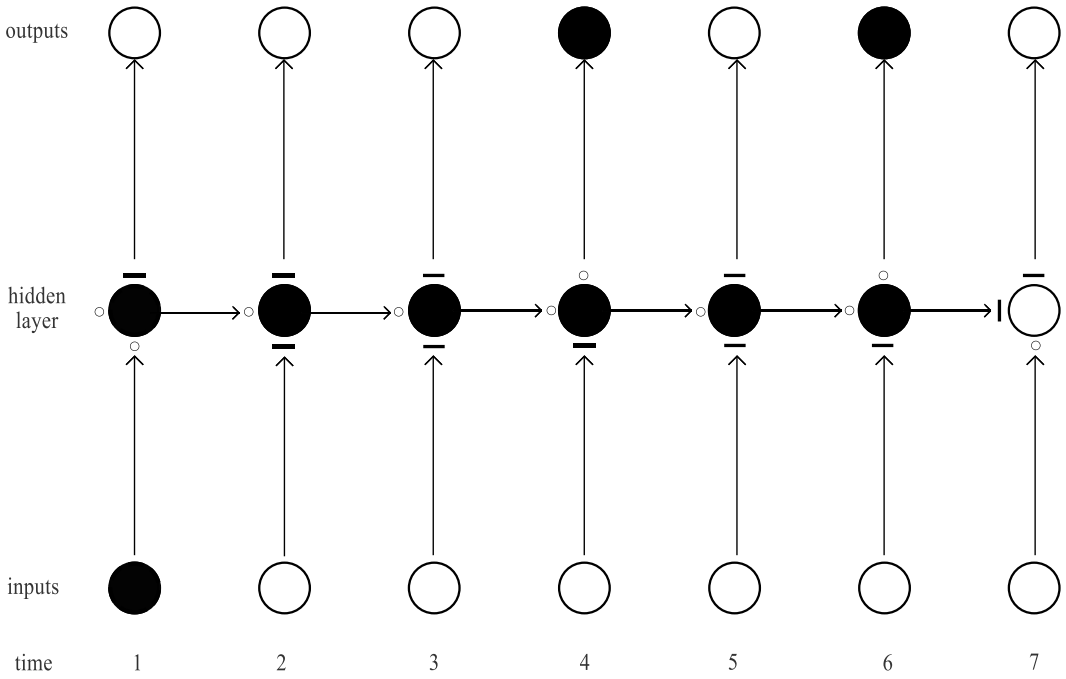

FIGURE 5.3: Preserving the input signal through blocking (-) or allowing (O) the input signal, adapted from Figure 4.4 of Graves [87]

forget gate is implemented to *reset* the cell. When a cell state got irrelevant due to a change in problem domain, forget gate gradually resets the cell state instead of erroneous activations from the input gate attempting to inefficiently do so.

Another extension came in the form of *peephole connections* by Gers *et al.* [90]. With this addition, the gates can use the cell state to increase their sensitivity to *timining* in the input data. By allowing internal gates to inspect the cell state, they have shown improvements on non-linear tasks.

The finalized model with input, forget and output gates as well as the internal peephole connections was debuted in Graves & Schmidhuber [91]. In their expansive study comparing 8 LSTM variants over 15 years of CPU time, Greff *et al.* [92] named this model the "vanilla LSTM". This particular form of LSTM is commonly used [81].

The real valued vectors are denoted alongside their update time step $(\cdot)_t$ such that $t \in 1, \ldots, T$. Updates are performed on the input gate i_t, memory cell c_t, forget gate f_t and output gate o_t. We show the updates over weight matrices W_i, W_f, W_c and W_o via the use of recurrent weights R_i, R_f, R_c and R_o and bias vectors b_i, b_f, b_c and b_o. Peephole connections are shown using $p.$ for input, forget and output gates as p_i, p_f and

p_o respectively. Finally, the input is a sequence of data in the form of $(x_1, x_2, \ldots, x_T)$ and the model outputs real valued vector y_t at the time step t.

$$i^t = \sigma\left(W_i x^t + R_i y^{t-1} + p_i \odot c^{t-1} + b_i\right) \tag{5.1}$$

$$f^t = \sigma\left(W_f x^t + R_f y^{t-1} + p_f \odot c^{t-1} + b_f\right) \tag{5.2}$$

$$\overline{c^t} = W_c x^t + R_c y^{t-1} + b_c \tag{5.3}$$

$$c_t = \tanh\left(i^t \odot \overline{c^t} + f_t \odot c^{t-1}\right) \tag{5.4}$$

$$o^t = \sigma\left(W_o x^t + R_o y^{t-1} + p_o \odot c^t + b_o\right) \tag{5.5}$$

$$y^t = \tanh(o_t) \odot c^t \tag{5.6}$$

Where $\sigma(\cdot)$ is the *logistic sigmoid* function $\sigma(x) = \frac{1}{1+e^{-x}}$ and pointwise vector multiplication is denoted using $\odot$. The recurrence holds via the usage of signals from the previous time steps $(\cdot)^{t-1}$.

5.2. Siamese Long Short-Term Memory

Recently, Mueller & Thyagarajan [93] proposed a *siamese* model to solve sentence similarity problem. They suggest using two discrete LSTM units that accept input from two separate streams. They are denoted as LSTM_a and LSTM_b. The weights are shared between the networks such that $\text{LSTM}_a = \text{LSTM}_b$. The Manhattan distance is used as the measure of similarity between the sentences.

We extend the siamese network architecture by using bilingual word embeddings. Our learning objective encodes a sense to be represented across languages using bilingual word embeddings.

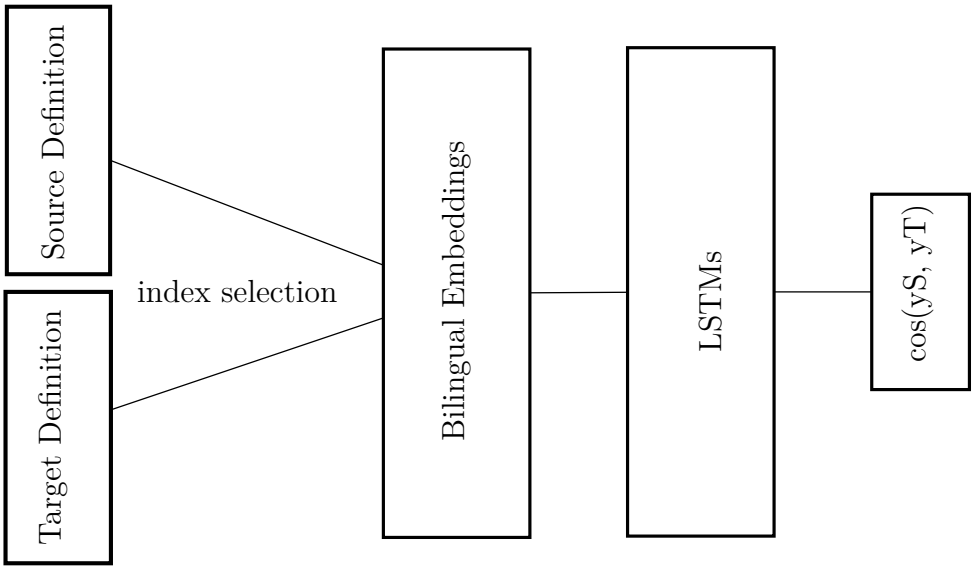

FIGURE 5.4: Overview of the siamese long short-term memory architecture used in the study

We used a similarity function that relies on cosine similarity over the outputs of the LSTM units.

$$\exp(\sum ||y^s \cdot y^t||_2) \tag{5.7}$$

Where y^s is the output of the LSTM that encodes the source definitions and y^t is the output of the LSTM that encodes the target definitions. We use adelta optimizer as suggested by **zeiler_adadelta_2012** and our loss function is the mean squared error. The LSTM weights are uniformly initialized [**glorot_understanding_2010**]. Furthermore, gradient clipping is employed in order to avoid the exploding gradient problem [**pascanu_difficulty_2012**]. We train our model using positive and negative samples. With the aligned golden corpora, we randomly shuffled half of the definitions and assigned them the label 0. The shuffle was done randomly since if we had chosen to only shuffle the latter half for instance, definitions that clump together might have related definitions or definitions from the same domain. So we tried to avoid assigning label 0 to related definitions.

The definitions are one hot encoded into a vector T. T is the indices of a definition's words and acts as a lookup for the embedding layer on top of the input layer. Any sentence that is longer than T will have words dropped. We have experimented with various values for T and found that more than twice the average word count across the

whole corpora work best. Following the input layer and the embedding layer, we have positioned the shared LSTM layer. The output of which is concatenated and fed into the mean squared error. We will report the accuracy of our model in Chapter 6.

6. Experiments and Evaluation

6.1. Preparing Wordnets

In order to run our experiments we need two sets of dictionary definitions from two different languages. Open Multilingual Wordnet [10] project hosts 34 wordnets with permissive licenses on their website.[1] We have investigated the available wordnets and found out that six of them included definitions, also known as glosses. Since we do not use any other information related to wordnets (like semantic relationships) only definitions were extracted into a plain text corpora. This intermediate corpora includes WordNet 3.0 synsets identifiers and the corresponding definitions in the target language. Natural Language Toolkit [94] provides an API for reading and retrieving English Princeton WordNet. Using the synsets identifiers, it is possible to retrieve the exact synset which comes attached with an unique definition. Finally, we have 6 aligned corpora for 6 wordnets we will run the experiments on. Alignment here refers to definitions that represent same synset across languages appearing on the same index or on the same line in their respective files. Throughout the chapter, we will base our evaluations on these alignments. The statistics and the 2 letter language codes that we will commonly use to denote the wordnets for the chapter is presented in Table 6.1.

Language Code	Language Name	Number of Definitions	Number of words	Average Words per Definition	Longest Definition
sq	Albanian	4681	54980	11.75	101
bg	Bulgarian	4959	63014	12.71	53
el	Greek	18136	203924	11.24	89
it	Italian	12688	93005	7.33	35
ro	Romanian	58754	586304	9.98	105
sl	Slovene	3144	39865	12.68	68

TABLE 6.1: Language codes and statistics for the target wordnets used in the thesis.

[1] http://compling.hss.ntu.edu.sg/omw

6.2. Preparing Word Embeddings

In Chapter 2, we have mentioned the recent popularization of pre-trained word embeddings. Initiated by word2vec[2], other sources for word embeddings are GloVe[3], fastText[4] and numberbatch[5]. Yet, word embeddings for languages other than English are scarce. fastText hosts word embeddings for 157 languages so we used them as our primary source [46]. Numberbatch is another approach for word embeddings and Speer *et al.* [21] host word embeddings for 78 different languages. Their embeddings are built on GloVe and word2vec and their model was explained in detail in Subsection 2.1.8. However, 10 of the available languages are presented as core languages with excellent support and 68 of them are provided as common languages which are only available with adequate support. Referring to Table 6.1, Italian is among the core languages and the rest are in the common languages group. We provide their models as additional comparison.

These embeddings are trained using Common Crawl and Wikipedia data, on billions of tokens. The lack of other embeddings models like GloVe in our study stems from the fact that downloading those huge corpora and training multilingual embeddings is a slow and expensive process.

The fastText embeddings include 2 million tokens out of the box. In order to increase the efficiency of the experiments, we have cut down the size of the embeddings into 1 million and 500 thousand while learning bilingual word embeddings. The vectors are sorted according to their corpus frequency so the uppermost lines where the most frequent tokens of the language reside are used. For the rest of the chapter, while presenting the evaluation results, the distinction between fastText vectors with 1 million tokens and 500 thousand tokens will be shown as *fastText 1M* and *fastText 500k* respectively.

Numberbatch embeddings are distributed in one large file and vectors are not in a set size across languages. Hence after parsing the file and extracting the embeddings into individual files, the number of word vectors we are left with are presented in Table 6.2.

[2] https://code.google.com/archive/p/word2vec
[3] https://nlp.stanford.edu/projects/glove
[4] https://fasttext.cc
[5] https://github.com/commonsense/conceptnet-numberbatch

Language Code	Number of Tokens
bg	20871
el	16926
en	417195
it	91829
ro	10874
sl	11458
sq	5512

TABLE 6.2: The number of word embeddings available in numberbatch

These embeddings are monolingual, so they are on separate arbitrary latent spaces. In order to represent them on the same latent space, we used VecMap [**artetxe__robust__2018**, **artetxe__generalizing__2018**, **artetxe__learning__2017**, 95].[6] According to **ruder__survey__201** bilingual or cross lingual embedding models optimize similar objectives and differences in performance is due to the data they are trained on. **glavas__how__2019** supports this suggestion and has empirically proven that common evaluation metrics like bilingual dictionary induction is not representative for the bilingual embedding's performance on downstream tasks. Hence our preference of VecMap is highly influenced by it's availability as an open source framework and it's ease of training.

Best performing VecMap model for learning bilingual embeddings that is available in the framework is *supervised alignment*. It requires a bilingual dictionary, otherwise known as aligned word pairs for two languages. We sourced our bilingual dictionary from Open Subtitles 2018[7] data as hosted by OPUS.[8] The dictionary can be sorted by the confidence score of the translation pair, which we did so that pairs with high confidence scores swam to the top. Topmost 25000 translation pairs were shuffled and split into training and testing examples for 6 languages. After supervised mapping of target language word embedding and English word embedding, we have 6 pairs of vectors that share the same latent space per pair, trained on 12500 word alignments.

Even though bilingual dictionary induction is not representative of a bilingual word embedding pair's performance on downstream tasks [**ruder__survey__2017**, **glavas__how__2019**], we include the evaluation results obtained from VecMap framework as a quick identifier for their performance on Table 6.3 for accuracy and Table 6.4

[6] https://github.com/artetxem/vecmap
[7] http://www.opensubtitles.org
[8] http://opus.nlpl.eu

Language Code	fastText 1M	fastText 500k	numberbatch
bg	33.61	35.17	51.97
el	37.37	39.58	30.35
it	58.20	59.28	50.37
ro	37.33	38.71	64.17
sl	21.42	22.91	74.74
sq	24.46	25.36	58.63

TABLE 6.3: Accuracy scores (in percentage) of the word embeddings aligned using VecMap

Language Code	fastText 1M	fastText 500k	numberbatch
bg	96.43	93.36	17.53
el	94.44	90.28	12.15
it	97.93	95.97	41.08
ro	97.06	94.91	16.4
sl	94.67	90.73	9.23
sq	83.59	80.92	9.51

TABLE 6.4: Coverage scores (in percentage) of the word embeddings aligned using VecMap

for coverage. Accuracy is the measure for correctly identifying the translation of a word given the test dictionary and coverage is the percentage of translation pairs that could be inducted. From the results, it is apparent that fastText embeddings have much better coverage due to vast data they were trained on. Yet, numberbatch exhibits better accuracy scores which might be a trade-off of their low coverage. Also note that Italian, which is the sole language in our experiment set which numberbatch claims to have full support for has significantly higher coverage score compared to rest of the languages. Nevertheless, bilingual dictionary induction results are not indicative of word embedding's performance on real life tasks.

6.3. Experiment Results

In this section, we will present the results of our experiments in the order they appeared in the main text of our study. Then, we will compare the approaches by presenting their results when ran on equivalent data.

6.3.1. Matching Results

To begin with, we evaluated linear assignment or matching approach using sentence embeddings and cosine similarity as the similarity measure between definitions. Using the graph analogy, nodes of the disjoints sets are dictionary definitions of two languages and edge weights between the nodes are the cosine similarity between the definitions across languages. Linear assignment algorithm proposed by Jonker & Volgenant [70] optimizes the overall cost of the bijection. This approach is presented in detail at Section 3.1. Since linear assignment is a one-to-one operation, we report precision at one or the percentage of definitions that are matched correctly.

The terms that do not have word embeddings available in the models are dropped from the definitions. If a definition falls shorter than 3 words after this operation, it is dropped from the experiment set completely. As a result, we report *cardinality*, the number of definitions available on two languages. Linear assignment using sentence embeddings using all of the available dictionary definitions per language is presented at Table 6.5.

Language Code	**Vector**	**Cardinality**	**Precision at one %**
bg	fastText 1M	4958	27.93
	fastText 500k	4958	28.52
	numberbatch	4896	12.99
el	fastText 1M	18117	31.15
	fastText 500k	18105	31.06
	numberbatch	17595	7.97
it	fastText 1M	12590	15.50
	fastText 500k	12585	15.73
	numberbatch	12563	20.63
sl	fastText 1M	3143	10.44
	fastText 500k	3143	11.42
	numberbatch	2955	4.70
sq	fastText 1M	4680	44.96
	fastText 500k	4680	44.81
	numberbatch	4639	16.06

TABLE 6.5: Evaluation results for linear assignment using sentence embeddings

Romanian wordnet definitions did not fit into the memory of the machine we tested on so we could not present the full definition matching results for Romanian wordnet.

In order to show a complete picture, the experiments are done on 3000 definition pairs per language which are presented in Table 6.6.

	Percentage of Correctly Matched Definitions		
Language Code	**fastText 1M**	**fastText 500k**	**Numberbatch**
bg	35.27	**36.00**	18.10
el	**36.13**	36.07	11.70
it	24.67	24.90	**32.07**
ro	36.43	**36.87**	18.73
sl	11.27	**11.40**	4.63
sq	39.43	**39.67**	19.03

TABLE 6.6: Definition matching evaluated on 3000 definition pairs

From Table 6.6, we can infer that fastText vectors truncated to 500 thousand tokens perform better overall while numberbatch is highly successful comparatively on the language it advertises full support for.

6.3.2. Google Translate Baseline

We have presented our method of translating the target wordnet definitions into English using Google Translate and running monolingual document retrieval on the corpora in Section 4.1. As mentioned before, we used the original English definitions extracted from English Princeton WordNet as the document collection to retrieve from and used the translated target wordnet definitions as queries to retrieve with. The experiments are done on 2000 definitions pairs for both languages.

The results are presented in Table 6.7. *tf-idf* weighted term document matrices and cosine similarity between definitions provide adequate results considering the simplicity of the approach. However, we should consider mean reciprocal rank. MRR penalizes the query according to the rank of the correct result, as the correct definition is retrieved on a lower rank, the MRR diminishes. The divide between the MRR score and the precision at one score indicates that machine translation introduces noise that keeps the correct definition from appearing among the top results heavily for the majority of the cases. We can draw attention to this fact by referring to Table 4.1. In the said table, we have shown examples of erronous sentence structures and usage of words that are semantically similar to their translation sources but does not make sense from a native speaker's standpoint. Such errors stem from machine translations and since

Google Translate is a proprietary software, we cannot draw conclusions from the inner workings of it, but only point at the results available to us.

Language	MRR %	Top 10 %	Top 1 %
bg	8.73	33.70	20.15
el	12.22	50.30	35.45
it	1.60	23.80	12.50
ro	5.07	49.40	36.40
sl	7.30	29.20	15.85
sq	5.56	48.95	38.35

TABLE 6.7: Evaluation results of Google Translate baseline

6.3.3. Cross Lingual Document Retrieval Results

We present the results of the two methods we have explained in detail at Section 4.2. Similar to how we built sentence embeddings, out of vocabulary words are handled simply by omitting them from the experiment data. Any individual definition that does not have at least 3 tokens are removed from the experiment set alongside its equivalent on the other language. We have chosen to remove 3 tokens arbitrarily, in order to remove potentially unrepresentative definitions as well as in order to keep the problem on the short document retrieval domain.

Cross lingual document retrieval is expensive in terms of memory requirements; in the study where the approach is proposed [20] as well as the open source implementation by the original author[9] which we modified for our study, the experiment set was kept at 500 documents on each language. So, after removing the definitions according to the guidelines above, 2000 definitions on both languages are chosen randomly as the experiment data.

We present mean reciprocal scores in Table 6.8 and precision at one scores in Table 6.9.

6.3.4. Performance Comparison

Balikas *et al.* [20] suggested entropic regularized version of word mover's distance. Their claim was that the smoothed out matrix can be solved more efficiently using the Sinkhorn-Knopp algorithm [**sinkhorn_concerning_1967**]. Since the original

[9] https://github.com/balikasg/WassersteinRetrieval

Language Code	**Vector**	**WMD tfidf**	**Sinkhorn tfidf**	**WMD tf**	**Sinkhorn tf**
bg	1M fastText	50.83	**51.85**	41.81	42.76
	500k fastText	51.15	**52.24**	42.18	43.19
	Numberbatch	**25.32**	24.97	18.55	18.09
el	1M fastText	47.82	**48.67**	40.74	41.42
	500k fastText	47.45	**48.45**	40.53	41.39
	Numberbatch	19.92	**20.06**	14.71	14.94
it	1M fastText	40.24	**40.60**	31.98	32.07
	500k fastText	40.27	**40.49**	32.11	32.21
	Numberbatch	42.72	**42.77**	35.11	35.12
ro	1M fastText	51.30	**51.95**	44.20	45.06
	500k fastText	51.28	**52.37**	43.85	45.14
	Numberbatch	27.68	**27.70**	21.57	21.86
sl	1M fastText	26.22	**26.38**	23.43	23.97
	500k fastText	26.12	**26.31**	23.47	24.03
	Numberbatch	9.35	**9.46**	6.35	6.26
sq	1M fastText	**65.66**	63.97	56.47	56.94
	500k fastText	**65.61**	64.42	56.61	57.05
	Numberbatch	**31.07**	31.06	24.31	24.74

TABLE 6.8: Mean reciprocal rank scores of cross lingual pseudo document retrieval approaches using word mover's distance and sinkhorn

Language Code	Vector	**WMD tfidf**	**Sinkhorn tfidf**	**WMD tf**	**Sinkhorn tf**
bg	1M fastText	41.50	**42.65**	33.95	34.60
	500k fastText	41.90	**43.00**	34.15	34.80
	Numberbatch	**17.45**	17.15	12.30	11.35
el	1M fastText	39.05	**40.15**	32.85	33.40
	500k fastText	38.45	**39.95**	32.55	33.55
	Numberbatch	13.15	**13.40**	9.95	10.00
it	1M fastText	31.15	**31.45**	23.50	23.40
	500k fastText	31.15	**31.30**	23.65	23.50
	Numberbatch	33.30	**33.35**	26.70	26.80
ro	1M fastText	**41.60**	41.50	35.90	35.60
	500k fastText	41.65	**42.20**	35.50	35.75
	Numberbatch	**19.85**	19.80	15.70	16.25
sl	1M fastText	17.80	**18.05**	15.10	15.60
	500k fastText	17.80	**17.95**	15.15	15.80
	Numberbatch	4.80	**5.00**	2.90	2.85
sq	1M fastText	**59.15**	56.40	48.55	49.30
	500k fastText	**58.70**	56.85	48.80	49.25
	Numberbatch	**23.55**	23.30	17.80	18.35

TABLE 6.9: Precision at one percentage scores for cross lingual pseudo document retrieval using word mover's distance and sinkhorn.

study did not report on timing performance results, we have timed our experiments and aligned our 6 corpora using different number of definition pairs. After completing the experiments, only timing information was relevant so we averaged the results over the experiments that used the same number of instances. We cannot iterate on the experiment using increasing number of definition pairs since the number of definitions available within each language's wordnet is restrictive. So we ran the timing study for 100, 200, 500 and 1000 definition pairs. To our surprise, there were next to no differences in terms of MRR or precision at one between Sinkhorn and Word Mover's Distance when the other variables were fixed. Yet, using Sinkhorn algorithm as suggested by Balikas *et al.* results in an average slowdown of 2.52 regarding our experiments. We present Figure 6.1 to compare the run times of two approaches that perform comparatively.

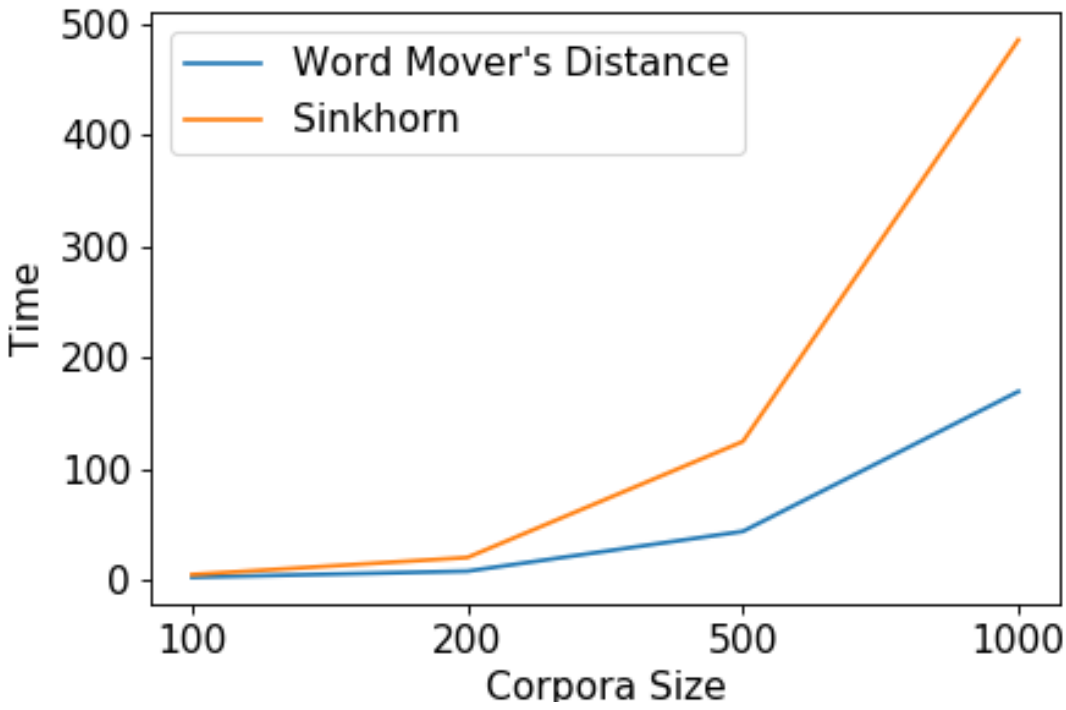

FIGURE 6.1: Timing comparison between word mover's distance and Sinkhorn algorithm

6.3.5. Supervised Alignment Results

In order to train our supervised dictionary definition encoder, we used Keras [96]. All the available definitions were used for this task so refer to Table 6.1 for the details on the size of the experiment data. Romanian wordnet includes the highest number of definitions, followed by Greek and Italian. Since the task is supervised, we would expect the highest accuracy from those wordnets.

We trained our models over 200 epochs but the accuracy and training loss plateau after 50 epochs so we only report up to 50^{th} epoch on our figures to reflect that. Due to highest coverage of tokens, we ran the supervised experiments using fastText embeddings, truncated to 1 million tokens. Learning rate of the model initialized on 1 and adapted dynamically. The plateau were hit as the learning rate adapted to 0.01. The input for the network was chosen as 25 words. Definitions that were longer than 25 words were truncated. The dimensions of the LSTM layer was selected as 100 which encoded fastText embeddings on 300 dimensions.

We present the training and *validation* accuracy results on Figure 6.2 The training and validation *loss* are presented in Table 6.3

Overall, we achieved the best validation accuracy, the accuracy on definitions that were never seen in training data with the language that has the highest number of training data avaiable, Romanian. After 15 epochs of training, accuracy of 65% is achieved and the accuracy fluctuates around 63% after 30^{th} epoch. Surprisingly, the encoder's performance was comparable among Bulgarian and Italian even though the latter has less than half the data available to train than Italian.

Language Code	Dictionary Size	Accuracy After Plateau
bg	4959	0.56
el	18136	0.60
it	12688	0.56
ro	58754	0.64
sl	3144	0.59
sq	4681	0.53

TABLE 6.10: The relation between the validation accuracy and the number of data points

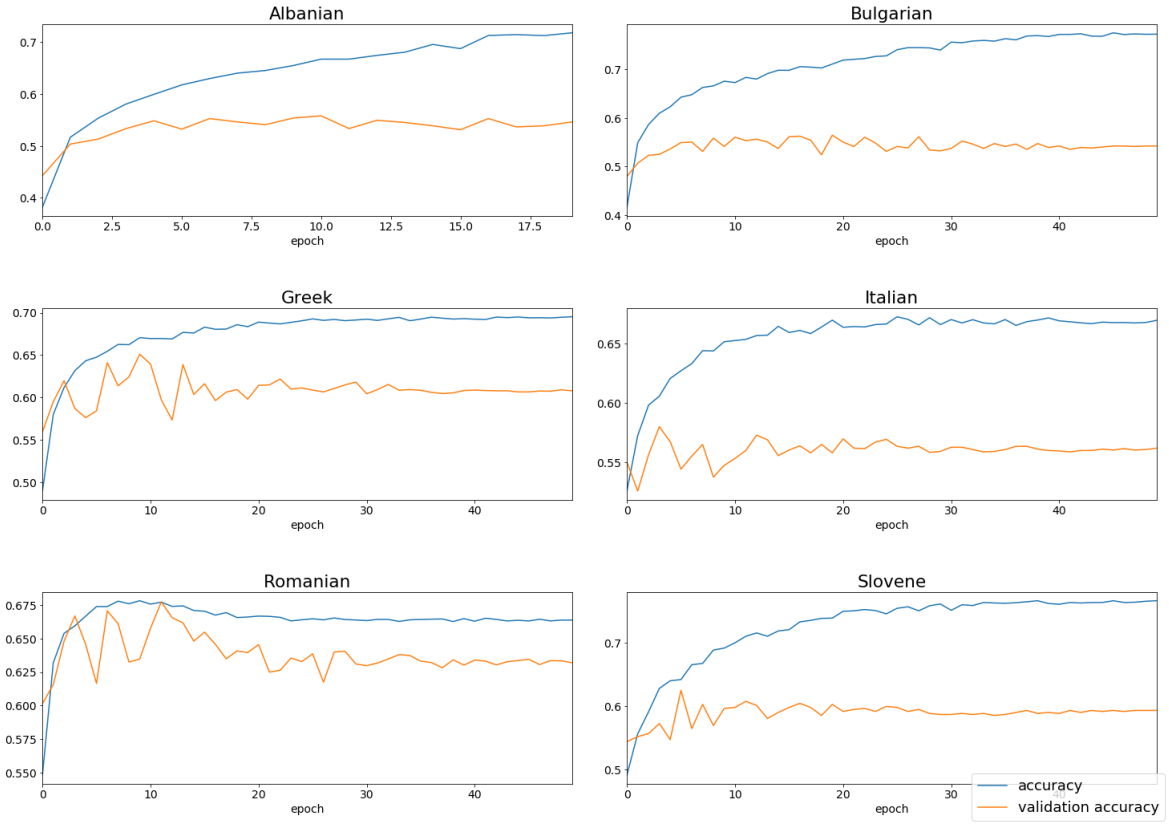

FIGURE 6.2: Accuracy of the supervised encoder on 6 wordnet corpora

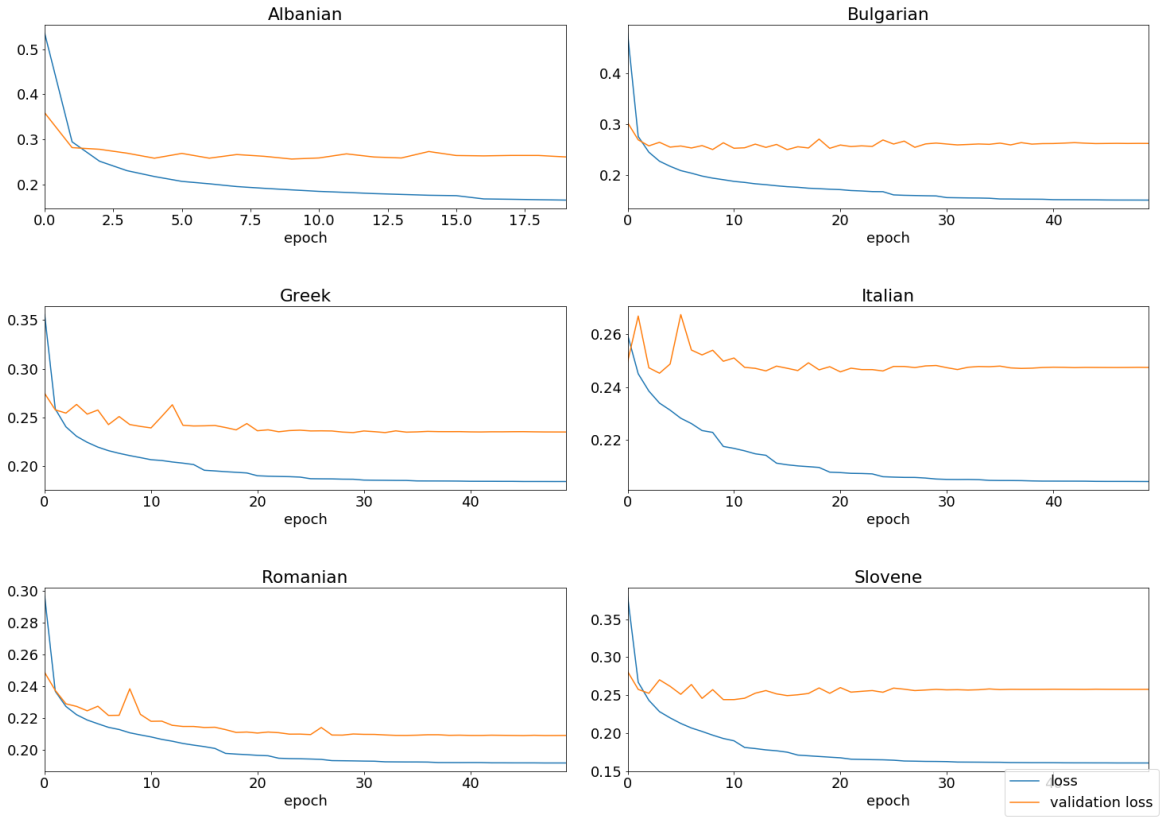

FIGURE 6.3: Loss of the supervised encoder on 6 wordnet corpora

6.4. Investigating Word Embedding Sources

We also experimented with *fastText* embeddings that were trained using the data available to us. The research question we are aiming to answer is if pre-trained embeddings are better than embeddings trained on the experiment data. Since Romanian wordnet has the most data available and word embedding's distributional models perform better as they learn from more distributional information, we have trained Romanian embeddings on the Romanian wordnet definitions. English fastText embeddings are trained on English Princeton WordNet definitions that were in the experiment set of Romaian wordnet. Then we mapped the embeddings to the same latent space using supervised VecMap and ran cross lingual document retrieval experiments using word mover's distance and Sinkhorn. Compared to MRR score of 51.30 for the word mover's distance and 51.95 with the Sinkhorn algorithm, using word embeddings trained using available data resulted in an MRR score of 28.69 for word mover's distance and 28.6 for Sinkhorn. Considering the drop in performance, we did not repeat the experiments for other language's corpora.

6.5. Comparative Analysis

In this section, we will compare the matching and the retrieval approaches by exploring the usage of metrics that were discussed on opposite sections. In other words, we *retrieved* dictionary definitions on the basis of cosine similarity between sentence embeddings of the dictionary definitions and *matched* dictionary definitions using word mover's and Sinkhorn distances as the edge weights of the bipartite graph.

In order to draw accurate conclusions, both matching and retrieval approaches were run on two sets constrained to 2000 dictionary definitions. Only fastText vectors that were truncated to most frequent 500 thousand (or fastText 500k in the context of this chapter) is used. Further, we only report *precision at one* or the percentage of correctly matched definitions in order to compare retrieval and matching approaches fairly. For cross lingual pseudo document retrieval, word mover's distance and Sinkhorn algorithms that use term definition matrices that were weighted using term frequency (Refer to Table 6.9) consistently perform worse than *tf-idf* weighted variants so they are not included in the comparisons.

We have split the presentation in two due to size constraints. Table 6.11 is the overview of the retrieval approaches. Using a metric to represent dictionary definitions and a similarity measure between them, given a query definition from target language, English Princeton WordNet definitions are ranked according to the similarity metric. The evaluation is the percentage of queries where English definition with the same index as the query is ranked on top. Balikas *et al.* [20] proposed using word mover's and Sinkhorn distances in order to retrieve documents in a different language than the query. Their state of the art model achieved the highest precision among the retrieval approaches we studied.

Using sentence embeddings in the context of matching is discussed in detail at Section 3.1. Extending said approach using word mover's distance or Sinkhorn distance is trivial. First, both metrics are used to populate a definition to definition distance matrix where $w_{i,j}$ is the distance calculated by either algorithm, between a term i of source language and a term j of the target language. Jonker-Volgenant algorithm is run in order to find the assignment where the overall sum of the distances is minimized. This approach is evaluated on the percentage of matched definitions with the same index. Table 6.12 presents the evaluation of matching approaches.

Looking a Table 6.12, **Sinkhorn** distance is the best performing metric. On Table 6.13, best precision at one percentage scores for retrieval and matching approaches are presented. Apart from the retrieval using word mover's distance on English and Albanian dictionary defintions, every score belongs to a metric that used Sinkhorn distance. Furthermore, matching has a clear advantage over the retrieval approaches.

6.6. Case Study

To observe the performance of dictionary alignment on a real life scenario we acquired a proprietary Turkish dictionary granted solely for research purposes. After parsing, 67351 headwords spanning 93062 definitions are extracted. Against 117000 synsets (that correspond to unique definitions) of WordNet, the size of the problem is not feasible due to memory restrictions of the pseudo document retrieval approach. We have tried to overcome it by running the experiment on only nouns but the issue persisted. As a result, as suggested by Khodak *et al.* [60], we constrained our scope to a list of core WordNet synsets. Open Multilingual Wordnet hosts[10] a list that denotes

[10] http://compling.hss.ntu.edu.sg/omw/wn30-core-synsets.tab

	Precision at one %			
Language Code	**WMD tfidf**	**Sinkhorn tfidf**	**Sentence Embedding**	**Google Translate Baseline**
bg	41.90	**43.00**	8.60	20.15
el	38.45	**39.95**	12.40	35.45
it	31.15	**31.30**	10.45	12.50
ro	41.65	**42.20**	14.70	36.40
sl	17.80	**17.95**	6.05	15.85
sq	**58.70**	56.85	10.65	38.35

TABLE 6.11: Comparison of the **retrieval** approaches presented in the study

	Precision at one %		
Language Code	**WMD tfidf**	**Sinkhorn tfidf**	**Sentence Embedding**
bg	49.95	**51.35**	40.75
el	65.65	**66.00**	37.70
it	39.45	**39.50**	28.25
ro	67.60	**68.20**	39.45
sl	28.16	**30.08**	15.05
sq	79.55	**79.65**	54.15

TABLE 6.12: Comparison of the **matching** approaches presented in the study

	Precision at one %	
Language Code	**Retrieval**	**Matching**
bg	43.00	51.35
el	39.95	66.00
it	31.30	39.50
ro	42.20	68.20
sl	17.95	30.08
sq	56.85	79.65

TABLE 6.13: Direct comparison between best performing matching and retrieval approaches

4961 WordNet identifiers in the form of offset and part of speech that is compatible with the nltk library, which was used to access to definitions of the identified synsets.[11] The list has been prepared by Boyd-Graber *et al.* [57] with the help of human evaluators by selecting salient synsets from a list of frequent words. Using a set of core WordNet synsets allowed us to pick a problem domain that can be tackled.

As further suggested by Khodak *et al.* [60], the identifiers for verbs and adjectives are deleted leaving only nouns. The final experiment set for Turkish dictionary definitions is prepared by translating the lemmas of the core WordNet synsets to Turkish and using the resulting list of lemmas to query the headwords of the Turkish dictionary. Using this method, we obtained 601 Turkish definitions. After removing the adjectives and verbs, 3280 WordNet definitions formed the definitions to retrieve against.

The approach for the case study is the word mover's distance using *tf-idf* weights, ran on fastText embeddings prepared using supervised VecMap. The bilingual dictionary provided by OpenSubtitles is used in order to map Turkish and English fastText embeddings.

While preparing the corpora for the pseudo document retrieval, 101 Turkish definitions are dropped due to them having no words to be represented by fastText embeddings while only 3 English definitions had to be omitted. Then, pseudo document retrieval is run over 501 Turkish definitions and 3277 English definitions.

In order to report on the performance for this task, we asked people to volunteer on scoring the resulting definition pairs. 100 definition pairs are chosen randomly among the 601 Turkish-English pairs and presented online for human annotators to score. We reached out to undergraduate students of TED University. The proficiency in English is required for the institution, so the volunteers should have an adequate grasp on the task.

The scale we presented included 3 scores. A score of "1" denoted that two definition pairs are completely unrelated, a score of "2" was asked if the pair of definitions are related and the score of "3" should be given for pairs that completely entail each other. The participants did not fill out every pair of definitions and 2 participants had to be omitted since they simply scored 1 or 3 for every definition pair respectively. At the end, we achieved 10.26 answers for each definition pair. Fleiss' Kappa measure [97]

[11] http://www.nltk.org/

is employed in order to measure the reliability of the given answers. The answer set scored $\kappa = 0.35$.

Percentage of Definitions		
Unrelated	**Related**	**Entails**
49.61	25.93	24.46

TABLE 6.14: Results of the case study; percentage of definitions that were agreed on by human annotators

According to the human referees, 24.46% of definitions completely entails each other while another 25.94% are related. However, volunteers marked another 49.61% of the definitions as unrelated. In Appendix A, we present the 100 randomly selected pairs of English definitions that were retrieved as the top result against the respective Turkish query.

7. Conclusion

In this study, we set out to investigate the feasibility of representing senses using their dictionary definitions. Along the way, we used document retrieval, linear programming and neural networks to answer the issue on as many angles as possible. The grand aim of the study was to compare the approaches that we had identified for the task. To our best knowledge, a comparable study where the dictionary alignment approaches were reported on the basis of their performance is not available so we had to anchor the study to itself. At the end of the day, we can make justified comparisons.

The monolingual retrieval using *tf-idf* weights and cosine similarity measure was chosen as a baseline because it is the most greedy approach available. If dictionary generation could be solved by automatic machine translation, this thesis would not take hold. The results presented in Chapter 6 prove so.

The matching algorithm is interesting. Moving on with our greedy connotation, for a task like dictionary alignment, assigning a sense to a definition that is closest to it by some distance metric might leave another definition with less than an ideal match later down the line. Matching ensures that the *closest* metric in between definitions holds not just for individual definitions but for the whole corpora. We can refer to Figure 3.1 to illustrate this point.

We have mentioned the lexical gap problem in Chapter 2 where some senses do not have equivalences in the target language. Recently, Bolukbasi *et al.* [98] reported on gender biases of word embedding models which numberbatch embeddings responded with so called de-biased embeddings, eliminating it from their models almost completely. Considering the most common type of lexical gap arises from languages with grammatical gender, possible effect of this on the matching approach is left for future work.

Overall, matching approach consistently shown the best performance across the board, supporting our hypothesis that one-to-one matching two sets of dictionary definitions would result in superior performance. We have also proven our justification behind the choice of the particular embedding model and the fact that conventional evaluation of word embeddings might not translate to downstream tasks. Numberbatch has scored first place on SemEval-2017 Task 2 [99], on multilingual word similarity task. Yet, against fastText embeddings, their model performed worse with the exception of Italian. Italian is a *core language* for numberbatch, where they claim full support. It

is also the language where numberbatch consistently outperformed fastText embeddings. We have set out to investigate the effect of particular choices like this for the dictionary alignment task. It can be reported with confidence that the advantage of one embedding model over another is not clear cut and should be investigated further.

With the supervised long short-term memory approach, we have observed that not only it is possible to represent senses using their dictionary definitions but also the metric of *representing* the same sense can be learned. The data required for obtaining any good performance should be noted and experimenting on diverse data should be left for future work.

The crucial shortcoming is the data requirements we have. On one hand, any type of description that represent a sense can be aligned not with just WordNet but any dictionary. Projects like BabelNet[1] or ConceptNet are creating semantic databases of their own while WordNet is on version 3.0, still online well after 20 years. Natural language processing research relies on external sources of information and the pre-annotated nature of these resources will always find a use. Working towards automatically extending them creates more opportunities for sprawling research later down the line.

Our main contribution in this study is the empirical comparison of alignment and retrieval approaches. We have hypothesized that aligning definitions one-to-one instead of greedily assigning each definition to it's closest counterpart will perform better. Our intuition behind the hypothesis is that dictionaries include discrete senses. Once a pair of definitions is matched, continuing to align further senses to any of the definitions can only deteriorate the performance. The results we have presented in Section 6.6 confirms our hypothesis. Matching approaches outperformed retrieval approaches on any language set. Including 6 different languages and observing the performance differences on all of them further confirms that by using the power of word embeddings, our finding are as language agnostic as possible. Our final conclusion is that the state of the art approach Sinkhorn distance [20] between term document representation outperformed sentence embeddings that were proposed specifically for short text representation. Further studies in the field can take this finding into account in their models.

[1] https://babelnet.org

7.1. Future Work

Throughout the thesis, English was always the centrepiece of the experiments. The wordnets were evaluated by their alignment towards the first and the most comprehensive, WordNet. The word embeddings were mapped to share a latent space with English word embeddings. As we have mentioned in Chapter 2, ideas like Inter-Lingual Index offer ways to bypass the English as a hub language. As an immediate future work, alignments that do not use English nor English Princeton WordNet can be investigated. Culturally or syntactically closer languages can be bridges more easily than distant yet abundant English.

Recent transfer learning models like BERT [100] offer a novel way to overcome the fundamental shortcoming with the supervised encoder we presented; the model performs in accordance with the available data and requires aligned data to function in the first place. Transfer learning inspires approaches like encoding the metric for representing the same sense in n languages after which the model is ready to predict on $n + 1^{th}$ language. Very recently, Jawanpuria *et al.* [101] proposed a VecMap like framework for convenient alignment of word embeddings. To our interest, the framework can map *multilingual* embeddings on a *shared space*. With a potential synset discovery approach like the one proposed by **ruiz-casado_automatic_2005** where possible sense definitions are found and validated using supervised learning will be investigated next using the novel ideas as inspiration.

Bibliography

1. *A Practical Guide to Lexicography* (ed van Sterkenburg, P.) *Terminology and Lexicography Research and Practice* **6.** OCLC: 249659375 (Benjamins, Amsterdam, 2003). 459 pp. ISBN: 978-90-272-2329-6 978-90-272-2330-2 978-1-58811-380-1 978-1-58811-381-8.
2. Uzun, E. N. Modern Dilbilim Bulguları Işığında Türkçe Sözlüğe Bir Bakış. *Çukurova Üniversitesi Türkoloji Araştırmaları Merkezi* (2005).
3. İbrahim USTA, H. Türkçe Sözlük Hazırlamada Yöntem Sorunları. *Ankara Üniversitesi Dil ve Tarih-Coğrafya Fakültesi Dergisi,* 223–242 (Jan. 1, 2006).
4. Uzun, L. 1945'TEN BU YANA TÜRKÇE SÖZLÜKLER. *KEBİKEÇ İnsan Bilimleri İçin Kaynak Araştırmaları Dergisi,* 53–57. ISSN: 1300-2864 (1999).
5. Kendall, J. *The Forgotten Founding Father: Noah Webster's Obsession and the Creation of an American Culture* 1st Edition. 368 pp. ISBN: 0-399-15699-2 (G.P. Putnam's Sons, Apr. 14, 2011).
6. Fellbaum, C. *WordNet : An Electronic Lexical Database* ISBN: 978-0-262-27255-1 (MIT Press, 1998).
7. Miller, G. A. Nouns in WordNet: A Lexical Inheritance System. *International Journal of Lexicography* **3,** 245–264. ISSN: 0950-3846. https://academic.oup.com/ijl/article/3/4/245/923281 (2019) (Dec. 1, 1990).
8. Winston, M. E., Chaffin, R. & Herrmann, D. A Taxonomy of Part-Whole Relations. *Cognitive Science* **11,** 417–444. ISSN: 1551-6709. https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1104_2 (2019) (1987).
9. Bond, F. & Paik, K. A Survey of WordNets and Their Licenses. *GWC 2012 6th International Global Wordnet Conference* **8,** 64 (Jan. 1, 2012).
10. Bond, F. & Foster, R. *Linking and Extending an Open Multilingual Wordnet* in. ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference. **1** (Aug. 1, 2013), 1352–1362.
11. Ruci, E. *On the Current State of Albanet and Related Applications* (Technical report, University of Vlora.(http://fjalnet. com …, 2008).
12. Simov, K. I. & Osenova, P. *Constructing of an Ontology-Based Lexicon for Bulgarian.* in *LREC* (Citeseer, 2010).
13. Stamou, S., Nenadic, G. & Christodoulakis, D. *Exploring Balkanet Shared Ontology for Multilingual Conceptual Indexing.* in *LREC* (2004).
14. Pianta, E., Bentivogli, L. & Girardi, C. MultiWordNet: Developing an Aligned Multilingual Database (Jan. 1, 2002).

15. Fišer, D., Novak, J. & Erjavec, T. *sloWNet 3.0: Development, Extension and Cleaning* in *Proceedings of 6th International Global Wordnet Conference (GWC 2012)* (2012), 113–117.
16. Tufiş, D., Ion, R., Bozianu, L., Ceauşu, A. & Ştefănescu, D. *Romanian Wordnet: Current State, New Applications and Prospects* in *Proceedings of 4th Global WordNet Conference, GWC* (2008), 441–452.
17. Somers, J. *You're Probably Using the Wrong Dictionary* http://jsomers.net/blog/dictionary (2019).
18. Almeida, F. & Xexéo, G. Word Embeddings: A Survey. arXiv: 1901.09069 [cs, stat]. http://arxiv.org/abs/1901.09069 (2019) (Jan. 25, 2019).
19. Collobert, R. & Weston, J. *A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning* in. Proceedings of the 25th International Conference on Machine Learning (ACM, May 7, 2008), 160–167. ISBN: 978-1-60558-205-4. http://dl.acm.org/citation.cfm?id=1390156.1390177 (2019).
20. Balikas, G., Laclau, C., Redko, I. & Amini, M.-R. *Cross-Lingual Document Retrieval Using Regularized Wasserstein Distance* in *Proceedings of the 40th European Conference ECIR Conference on Information Retrieval, ECIR 2018, Grenoble, France, March 26-29, 2018* (2018).
21. Speer, R., Chin, J. & Havasi, C. *ConceptNet 5.5: An Open Multilingual Graph of General Knowledge* in. AAAI Conference on Artificial Intelligence (2017), 4444–4451. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972.
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. arXiv: 1310.4546 [cs, stat]. http://arxiv.org/abs/1310.4546 (2019) (Oct. 16, 2013).
23. Pennington, J., Socher, R. & Manning, C. *Glove: Global Vectors for Word Representation* in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), 1532–1543.
24. Harris, Z. S. Distributional Structure. *WORD* **10,** 146–162. ISSN: 0043-7956. https://doi.org/10.1080/00437956.1954.11659520 (2019) (Aug. 1, 1954).
25. Firth, J. R. A Synopsis of Linguistic Theory 1930–1955. *Studies in linguistic analysis* **Special volume of the Phiological Society,** 11 (1957).
26. Osgood, C. E., Suci, G. J. & Tannenbaum, P. H. *The Measurement of Meaning* 358 pp. ISBN: 978-0-252-74539-3 (University of Illinois Press, 1957).

27. Lund, K. & Burgess, C. Producing High-Dimensional Semantic Spaces from Lexical Co-Occurrence. *Behavior Research Methods, Instruments, & Computers* **28,** 203–208. ISSN: 1532-5970. https://doi.org/10.3758/BF03204766 (2019) (June 1, 1996).
28. Salton, G., Wong, A. & Yang, C. S. A Vector Space Model for Automatic Indexing. *Commun. ACM* **18,** 613–620. ISSN: 0001-0782. http://doi.acm.org/10.1145/361219.361220 (2019) (Nov. 1975).
29. Turney, P. D. & Pantel, P. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research* **37,** 141–188. ISSN: 1076-9757. arXiv: 1003.1141. http://arxiv.org/abs/1003.1141 (2018) (Feb. 27, 2010).
30. Jones, K. S. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation* **28,** 11–21 (1972).
31. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. Indexing by Latent Semantic Analysis. *Journal of the American society for information science* **41,** 391–407 (1990).
32. Levy, O., Goldberg, Y. & Dagan, I. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics* **3,** 211–225 (Dec. 1, 2015).
33. Church, K. W. & Hanks, P. Word Association Norms, Mutual Information, and Lexicography. *Comput. Linguist.* **16,** 22–29. ISSN: 0891-2017. http://dl.acm.org/citation.cfm?id=89086.89095 (2019) (Mar. 1990).
34. Forsythe, G. E., Malcolm, M. A. & Moler, C. B. *Computer Methods for Mathematical Computations* (Prentice-hall Englewood Cliffs, NJ, 1977).
35. Landauer, T. K. & Dumais, S. T. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological review* **104,** 211 (1997).
36. Schütze, H. *Dimensions of Meaning* in *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing* Minneapolis, Minnesota, USA (IEEE Computer Society Press, Los Alamitos, CA, USA, 1992), 787–796. ISBN: 978-0-8186-2630-2. http://dl.acm.org/citation.cfm?id=147877.148132 (2019).
37. Xu, W. & Rudnicky, A. *Can Artificial Neural Networks Learn Language Models?* in *Sixth International Conference on Spoken Language Processing* (2000).
38. P. Turian, J., Ratinov, L.-A. & Bengio, Y. *Word Representations: A Simple and General Method for Semi-Supervised Learning.* in. Proceedings of the 48th

Annual Meeting of the Association for Computational Linguistics. **2010** (Jan. 1, 2010), 384–394.

39. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient Estimation of Word Representations in Vector Space. arXiv: 1301.3781 [cs]. http://arxiv.org/abs/1301.3781 (Jan. 16, 2013).
40. Mikolov, T., Yih, W.-t. & Zweig, G. *Linguistic Regularities in Continuous Space Word Representations* in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2013), 746–751.
41. Bullinaria, J. A. & Levy, J. P. Extracting Semantic Representations from Word Co-Occurrence Statistics: A Computational Study. *Behavior Research Methods,* 510–526 (2007).
42. Baroni, M., Dinu, G. & Kruszewski, G. *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)* Baltimore, Maryland (Association for Computational Linguistics, June 2014), 238–247. https://www.aclweb.org/anthology/P14-1023.
43. Levy, O. & Goldberg, Y. *Neural Word Embedding As Implicit Matrix Factorization* in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* Montreal, Canada (MIT Press, Cambridge, MA, USA, 2014), 2177–2185. http://dl.acm.org/citation.cfm?id=2969033.2969070 (2019).
44. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C. & Joulin, A. *Advances in Pre-Training Distributed Word Representations* in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)* (2018).
45. Mnih, A. & Kavukcuoglu, K. in *Advances in Neural Information Processing Systems 26* (eds Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z. & Weinberger, K. Q.) 2265–2273 (Curran Associates, Inc., 2013). http://papers.nips.cc/paper/5165-learning-word-embeddings-efficiently-with-noise-contrastive-estimation.pdf.
46. Grave, E., Bojanowski, P., Gupta, P., Joulin, A. & Mikolov, T. Learning Word Vectors for 157 Languages. arXiv: 1802.06893 [cs]. http://arxiv.org/abs/1802.06893 (2019) (Feb. 19, 2018).

47. Anacleto, J. *et al. Can Common Sense Uncover Cultural Differences in Computer Applications?* in *Artificial Intelligence in Theory and Practice* (ed Bramer, M.) (Springer US, 2006), 1–10. ISBN: 978-0-387-34747-9.
48. Auer, S. *et al. DBpedia: A Nucleus for a Web of Open Data* in *The Semantic Web* (eds Aberer, K. *et al.*) (Springer Berlin Heidelberg, 2007), 722–735. ISBN: 978-3-540-76298-0.
49. Faruqui, M. & Dyer, C. *Improving Vector Space Word Representations Using Multilingual Correlation* in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics* (Association for Computational Linguistics, Gothenburg, Sweden, 2014), 462–471. http://aclweb.org/anthology/E14-1049 (2018).
50. Neale, S. *A Survey on Automatically-Constructed WordNets and Their Evaluation: Lexical and Word Embedding-Based Approaches* in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* Miyazaki, Japan (eds Calzolari, N. *et al.*) (European Language Resources Association (ELRA), May 7–12, 2018). ISBN: 979-10-95546-00-9.
51. Ercan, G. & Cicekli, I. Using Lexical Chains for Keyword Extraction. *Information Processing & Management* **43,** 1705–1714 (2007).
52. Banerjee, S. & Pedersen, T. *An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet* in *Computational Linguistics and Intelligent Text Processing* (ed Gelbukh, A.) (Springer Berlin Heidelberg, 2002), 136–145. ISBN: 978-3-540-45715-2.
53. Vossen, P. EUROWORDNET: A MULTILINGUAL DATABASE OF AUTONOMOUS AND LANGUAGE-SPECIFIC WORDNETS CONNECTED VIA AN INTER-LINGUALINDEX. *International Journal of Lexicography* **17,** 161–173. ISSN: 0950-3846. https://academic.oup.com/ijl/article/17/2/161/969685 (2019) (June 1, 2004).
54. Gonzalo, J., Verdejo, F., Peters, C. & Calzolari, N. in *EuroWordNet: A Multilingual Database with Lexical Semantic Networks* (ed Vossen, P.) 113–135 (Springer Netherlands, Dordrecht, 1998). ISBN: 978-94-017-1491-4. https://doi.org/10.1007/978-94-017-1491-4_5 (2019).
55. Kitamura, K. Cultural Untranslatability. *Translation Journal* **13** (2009).
56. Knight, K. & Luk, S. K. *Building a Large-Scale Knowledge Base for Machine Translation* in *In Proceedings of AAAI* (1994).

57. Boyd-Graber, J., Fellbaum, C., Osherson, D. & Schapire, R. *Adding Dense, Weighted Connections to WordNet* in *Proceedings of the Third International WordNet Conference* (Citeseer, 2006), 29–36.
58. Fiser, D. *Leveraging Parallel Corpora and Existing Wordnets for Automatic Construction of the Slovene Wordnet* in (Aug. 25, 2009), 359–368.
59. Lam, K. N., Tarouti, F. A. & Kalita, J. *Automatically Constructing Wordnet Synsets* in. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (June 2014), 106–111. https://aclweb.org/anthology/papers/P/P14/P14-2018/ (2019).
60. Khodak, M., Risteski, A., Fellbaum, C. & Arora, S. *Automated WordNet Construction Using Word Embeddings* in *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and Their Applications* (2017), 12–23.
61. Lesk, M. *Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone* in *Proceedings of the 5th Annual International Conference on Systems Documentation* (ACM, New York, NY, USA, 1986), 24–26. ISBN: 978-0-89791-224-2. http://doi.acm.org/10.1145/318723.318728.
62. Gordeev, D., Rey, A. & Shagarov, D. *Unsupervised Cross-Lingual Matching of Product Classifications* in *Proceedings of the 23rd Conference of Open Innovations Association FRUCT* Bologna, Italy (FRUCT Oy, 2018), 62:459–62:464. http://dl.acm.org/citation.cfm?id=3299905.3299967.
63. Le, Q. V. & Mikolov, T. Distributed Representations of Sentences and Documents. arXiv: 1405.4053 [cs]. http://arxiv.org/abs/1405.4053 (May 16, 2014).
64. Kiros, R. *et al.* Skip-Thought Vectors. arXiv: 1506.06726 [cs]. http://arxiv.org/abs/1506.06726 (2019) (June 22, 2015).
65. Wieting, J., Bansal, M., Gimpel, K. & Livescu, K. Towards Universal Paraphrastic Sentence Embeddings. arXiv: 1511.08198 [cs]. http://arxiv.org/abs/1511.08198 (2019) (Nov. 25, 2015).
66. Zhao, J., Lan, M. & Tian, J. *ECNU: Using Traditional Similarity Measurements and Word Embedding for Semantic Textual Similarity Estimation* in *SemEval@NAACL-HLT* (2015).
67. Šarić, F., Glavaš, G., Karan, M., Šnajder, J. & Bašić, B. D. *TakeLab: Systems for Measuring Semantic Text Similarity* in *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of*

the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation Montréal, Canada (Association for Computational Linguistics, Stroudsburg, PA, USA, 2012), 441–448. http://dl.acm.org/citation.cfm?id=2387636.2387708 (2019).

68. Edilson A. Corrêa, J., Marinho, V. & Borges dos Santos, L. NILC-USP at SemEval-2017 Task 4: A Multi-View Ensemble for Twitter Sentiment Analysis (Apr. 7, 2017).
69. Kuhn, H. W. The Hungarian Method for the Assignment Problem. *Naval research logistics quarterly* **2,** 83–97 (1955).
70. Jonker, R. & Volgenant, A. A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems. *Computing* **38,** 325–340. ISSN: 1436-5057. https://doi.org/10.1007/BF02278710 (2019) (Dec. 1, 1987).
71. Dijkstra, E. W. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* **1,** 269–271. ISSN: 0945-3245. https://doi.org/10.1007/BF01386390 (2019) (Dec. 1, 1959).
72. Ruder, S., Cotterell, R., Kementchedjhieva, Y. & Søgaard, A. A Discriminative Latent-Variable Model for Bilingual Lexicon Induction. arXiv: 1808.09334 [cs, stat]. http://arxiv.org/abs/1808.09334 (2019) (Aug. 28, 2018).
73. Bush, V. As We May Think. *The atlantic monthly* **176,** 101–108 (1945).
74. Singhal, A. Modern Information Retrieval: A Brief Overview. *Bulletin of the Ieee Computer Society Technical Committee on Data Engineering* **24,** 2001 (2001).
75. Luhn, H. P. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of research and development* **1,** 309–317 (1957).
76. Groves, M. & Mundt, K. Friend or Foe? Google Translate in Language for Academic Purposes. *English for Specific Purposes* **37,** 112–121. ISSN: 0889-4906. http://www.sciencedirect.com/science/article/pii/S088949061400060X (2019) (Jan. 1, 2015).
77. Manning, C. D., Raghavan, P. & Schütze, H. *Introduction to Information Retrieval* Reprinted. OCLC: 549201180. 482 pp. ISBN: 978-0-521-86571-5 (Cambridge Univ. Press, Cambridge, 2009).
78. Ruiz-Casado, M., Alfonseca, E. & Castells, P. *Automatic Assignment of Wikipedia Encyclopedic Entries to Wordnet Synsets* in *International Atlantic Web Intelligence Conference* (Springer, 2005), 380–386.

79. Salton, G. The State of Retrieval System Evaluation. *Information processing & management* **28,** 441–449 (1992).
80. Voorhees, E. M. *The TREC-8 Question Answering Track Report* in *In Proceedings of TREC-8* (1999), 77–82.
81. Sutskever, I., Vinyals, O. & Le, Q. V. Sequence to Sequence Learning with Neural Networks. arXiv: 1409.3215 [cs]. http://arxiv.org/abs/1409.3215 (2019) (Sept. 10, 2014).
82. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning Representations by Back-Propagating Errors. *Nature* **323,** 533. ISSN: 1476-4687. https://www.nature.com/articles/323533a0 (2019) (Oct. 1986).
83. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **9,** 1735–1780. ISSN: 0899-7667. http://dx.doi.org/10.1162/neco.1997.9.8.1735 (2019) (Nov. 1997).
84. Graves, A., Liwicki, M., Bunke, H., Schmidhuber, J. & Fernández, S. in *Advances in Neural Information Processing Systems 20* (eds Platt, J. C., Koller, D., Singer, Y. & Roweis, S. T.) 577–584 (Curran Associates, Inc., 2008). http://papers.nips.cc/paper/3213-unconstrained-on-line-handwriting-recognition-with-recurrent-neural-networks.pdf.
85. Graves, A. *et al.* A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31,** 855–868. ISSN: 0162-8828 (May 2009).
86. Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O. & Zaremba, W. Addressing the Rare Word Problem in Neural Machine Translation. arXiv: 1410.8206 [cs]. http://arxiv.org/abs/1410.8206 (2019) (Oct. 29, 2014).
87. Graves, A. in *Supervised Sequence Labelling with Recurrent Neural Networks* (ed Graves, A.) 37–45 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012). ISBN: 978-3-642-24797-2. https://doi.org/10.1007/978-3-642-24797-2_4 (2019).
88. Bengio, Y., Simard, P. & Frasconi, P. Learning Long-Term Dependencies with Gradient Descent Is Difficult. *IEEE Transactions on Neural Networks* **5,** 157–166. ISSN: 1045-9227 (Mar. 1994).
89. Gers, F. A., Schmidhuber, J. & Cummins, F. A. Learning to Forget: Continual Prediction with LSTM. *Neural Computation* **12,** 2451–2471 (2000).
90. Gers, F. A., Schraudolph, N. N. & Schmidhuber, J. Learning Precise Timing with Lstm Recurrent Networks. *J. Mach. Learn. Res.* **3,** 115–143. ISSN: 1532-4435. https://doi.org/10.1162/153244303768966139 (2019) (Mar. 2003).

91. Graves, A. & Schmidhuber, J. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks. IJCNN 2005* **18,** 602–610. ISSN: 0893-6080. http://www.sciencedirect.com/science/article/pii/S0893608005001206 (2019) (July 1, 2005).
92. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R. & Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems* **28,** 2222–2232. ISSN: 2162-237X, 2162-2388. arXiv: 1503.04069. http://arxiv.org/abs/1503.04069 (2019) (Oct. 2017).
93. Mueller, J. & Thyagarajan, A. *Siamese Recurrent Architectures for Learning Sentence Similarity* in *AAAI* (2016).
94. Bird, S., Klein, E. & Loper, E. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit* 506 pp. ISBN: 978-0-596-55571-9 ("O'Reilly Media, Inc.", June 12, 2009).
95. Artetxe, M., Labaka, G. & Agirre, E. *Learning Principled Bilingual Mappings of Word Embeddings While Preserving Monolingual Invariance* in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (2016), 2289–2294.
96. Chollet, F. *et al.* Keras. https://keras.io (2015).
97. Fleiss, J. L. Measuring Nominal Scale Agreement among Many Raters. *Psychological bulletin* **76,** 378 (1971).
98. Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V. & Kalai, A. Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings. arXiv: 1607.06520 [cs, stat]. http://arxiv.org/abs/1607.06520 (2019) (July 21, 2016).
99. Camacho-Collados, J., Pilehvar, M. T., Collier, N. & Navigli, R. *SemEval-2017 Task 2: Multilingual and Cross-Lingual Semantic Word Similarity* in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* Vancouver, Canada (Association for Computational Linguistics, Aug. 2017), 15–26. http://www.aclweb.org/anthology/S17-2002.
100. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv: 1810.04805 [cs]. http://arxiv.org/abs/1810.04805 (2019) (Oct. 10, 2018).
101. Jawanpuria, P., Balgovind, A., Kunchukuttan, A. & Mishra, B. Learning Multilingual Word Embeddings in Latent Metric Space: A Geometric Approach.

Transactions of the Association for Computational Linguistics **7,** 107–120. https://doi.org/10.1162/tacl_a_00257 (2019) (Mar. 1, 2019).

A. Case Study - Aligning a Turkish Dictionary and English Princeton WordNet

Turkish Definition	English WordNet Definition
birdenbire duyulan ağrı ya da türlü heyecanları anlatır	an absence of emotion or enthusiasm
meyve bisküvi vb ile yapılan bir ïngiliz tatlısı	a small hard fruit
yalıtım tecrit	a thin coating or layer
bulgu araz	a pattern of symptoms indicative of some disease
42195 m'lik en uzun yaya koşusu uzunkoşu	any long ditch cut in the ground
benzer olma durumu müşabehet	the quality of being similar
var olan bulunan	something that is lost
parmakların kapanmasıyla elin aldığı biçim	handwear covers the hand and wrist
zayıf olma durumu	any strong feeling
bir süreç içindeki durumlardan her biri basamak aşama rütbe mertebe	the effect of one thing or person on another
bir birimin bölündüğü eşit parçalardan birini ya da birkaçını anlatan sayı	a unit of spoken language larger than a phoneme
haşhaş sütünü toplamakta kullanılan kaşık	grass mowed and cured for use as fodder
bir şeyi yapmayı önceden isteyip düşünme maksat	an act of intending a volition that you intend to carry out
ïpekböceği kozaları çözülerek çıkarılan ve dokumacılıkta kullanılan çok ince esnek ve parlak tel	a very strong thick rope made of twisted hemp or steel wire

Continued on next page

Table A.1 – *Continued from previous page*

Turkish Definition	**English WordNet Definition**
anadolu'nun doğu ve kuzey bölgelerinde en çok erzurum yöresinde el ele tutuşarak oynanan bir oyun	game a players turn to take some action permitted by the rules of the game
başkalarından saklanan duyurulmayan saklı kalan mahrem	a consecrated place where sacred objects are kept
bir kimseye ya da bir şeye karşı belli tavır takınmak	a feeling of sympathy for someone or something
tarihsel gelişme içinde belirli bir üretim ilişkisinin belirli bir aşamasında bir arada yaşayan insanların tümü	a large number of things or people considered together
bir cismin herhangi bir yöndeki uzanımı	a relatively small granular particle of a substance
organizmada birtakım değişikliklerin ortaya çıkmasıyla fizyoloji görevlerinin bozulması durumu sayrılık maraz esenlik" karşıtı	an impairment of health or a condition of abnormal functioning
yolcu ve gezmenlere geceleme olanağı sağlamak bunun yanında yemek eğlence gibi türlü hizmetleri sunmak ereğiyle kurulmuş işletme	a building where travelers can pay for lodging and meals and other services
hükümdar ailesinden olan erkeklere verilen san	female of any member of the dog family
orduda yazı işleri ile uğraşan er ya da erbaş	a verbal or written request for assistance or employment or admission to a school
düşmanın gelmesi beklenen yollar üzerinde askeri önem taşıyan kentlerde geçit ve darboğazlarda güvenliği sağlamak için yapılan kalın duvarlı burçlu mazgallı yapı	an area within a building enclosed by walls and floor and ceiling

Continued on next page

Table A.1 – *Continued from previous page*

Turkish Definition	**English WordNet Definition**
ïçi su ya da hava dolu ufak kabartı ya da kürecik	water in small drops in the atmosphere blown from waves or thrown up by a waterfall
hava ya da herhangi bir akışkanı bir yerden başka bir yere basınç yoluyla aktarmaya yarayan makine	a vertical flue that provides a path through which smoke from a fire is carried away through the wall or roof of a building
başkalarınca bilinmesi sakıncalı görülen bir gerçeği saklamaktan vazgeçip açıklama söyleme bildirme	an acknowledgment of the truth of something
bir bilim sanat meslek dalıyla ya da bir konu ile ilgili özel ve belirli bir kavramı olan sözcük ıstılah	an occupation requiring special education especially in the liberal arts or sciences
bir kilidi açıp kapamak için kullanılan araç açar açkı	a fastener fitted to a door or drawer to keep it firmly closed
ïşletilmek için bir yere ödünç verilen paraya karşılık alınan kâr ürem işlenti nema	a fixed charge for borrowing money usually a percentage of the amount borrowed
yeryüzü parçası yerey yer toprak alan	sloping land especially the slope beside a body of water
yiyecek koymaya yarar az derin ve yayvan kap	metal or earthenware cooking vessel that is usually round and deep often has a handle and lid
yerleşik toplumsal düzeni köklü hızlı ve geniş kapsamlı olarak niteliksel değiştirme ve yeniden biçimlendirme eylemi inkılap	an extended social group having a distinctive cultural and economic organization
duygusal olma durumu	an unstable situation of extreme danger or difficulty

Continued on next page

Table A.1 – *Continued from previous page*

Turkish Definition	**English WordNet Definition**
özel bir bozukluğu belirleyen bir arada görülen tanıyı kolaylaştıran bulgu ve belirtilerin tümü	a pattern of symptoms indicative of some disease
dumanı ocaktan çekip havaya vermeye yarayan maden ya da tuğla yol	a vertical flue that provides a path through which smoke from a fire is carried away through the wall or roof of a building
bir canlının üstünü kaplayan ve onu dış etkilere karşı koruyan kendiliğinden oluşmuş sertçe bölüm kışır	the activity of protecting someone or something
bir olayın ilk duyurusu olan biten salık	following the first in an ordering or series
çoğunlukla kare ya da silindir biçimindeki yüksek yapı	a protective covering or structure
bir yazıya başka bir yazarın yazısından alınmış parça aktarma iktibas	the form in which a text especially a printed book is published
felsefeyle uğraşan ve felsefenin gelişmesine katkıda bulunan kimse felsefeci feylesof	a specialist in philosophy
bir vücudun ya da bir organın yapı öğelerinden birini oluşturan gözeler bütünü nesiç	part of an organism consisting of an aggregate of cells having a similar structure and function
fiziksel güç takat	the property of lacking physical or mental strength liability to failure under pressure or stress or strain
ilgisini çekmek önem vermek ya da bir şeyle ilgili kılmak	an act of help or assistance
askeri olmayan	a nonmilitary citizen
bir olayı gören izleyen kimse izleyici	someone who takes part in an activity

Continued on next page

Table A.1 – *Continued from previous page*

Turkish Definition	**English WordNet Definition**
yapıları ve ulaşım araçlarını tren vapur gibi aydınlatmak havalandırmak amacıyla yapılan çerçeve cam panjur perde gibi eklentilerle daha kullanışlı bir duruma getirilen açıklık	a vertical flue that provides a path through which smoke from a fire is carried away through the wall or roof of a building
ïnsanda ve omurgalılarda içinde beyin bulunan başın kemik bölümü	the bony skeleton of the head of vertebrates
özellikle sokakta ayağı korumak için giyilen iskarpin çizme kundura mokasen sandalet patik galoş gibi türleri olan ayak giyeceği pabuç	a shoe consisting of a sole fastened by straps to the foot
yüzeyi belirli uzunlukta bırakılmış hammadde lifleriyle kaplı parlak yumuşak kumaş	fabric woven from cotton fibers
delik yırtık ya da eski bir yeri uygun bir parça ile onarma kapatma	a piece of cloth used as decoration or to mend or cover a hole
ciltli ya da ciltsiz olarak bir araya getirilmiş basılı ya da yazılı kâğıt yaprakların tümü	one side of one leaf of a book or magazine or newspaper or letter etc or the written or pictorial matter it contains
kuvvetin bir cismi bir nokta ya da bir eksen yöresinde döndürme etkisini belirleyen vektör niceliği	the effect of one thing or person on another
bir evin en geniş bölümü	an outbuilding or part of a building for housing automobiles
bez tahta kâğıt gibi maddeler üzerine yapılmış yağlıboya suluboya pastel ya da karakalem resim	a three-dimensional work of plastic art
ïlgisiz olma durumu aldırmazlık alakasızlık	an unstable situation of extreme danger or difficulty

Continued on next page

Table A.1 – *Continued from previous page*

Turkish Definition	**English WordNet Definition**
sevgi ve bağlılık duyulan	an absence of emotion or enthusiasm
anahtar düğme gibi takılıp çıkarılabilen bir parça yardımıyla çalışan kimi zaman elektronik de olabilen kapatma aygıtı	electronic equipment consisting of a device providing access to a computer has a keyboard and display
evrende ya da düşüncede yer alan "yok" karşıtı bu sözcük hep yüklem olarak kullanılır ve üçüncü kişilerde koşaç almayabilir belirten olması için sonuna olan ortacı getirilir	something that should remain hidden from others especially information that is not to be passed on
bir yapının dışarıya doğru çıkmış çevresi duvar ya da parmaklıkla çevrili bölümü	a movable barrier in a fence or wall
evin ya da herhangi bir yapının oturmak çalışmak yatmak gibi işlere yarayan banyo salon giriş vb dışında kalan bir ya da birden fazla çıkışı olan bölmesi göz	a structure taller than its diameter can stand alone or be attached to a larger building
kurulanmaya yarar havlı bez	white goods or clothing made with linen cloth
kamuyla ilgili işlerin yürütülmesi için gerekli gelirleri ve harcanan paraları düzenleyen kuralların tümü	the management of money and credit and banking and investments
bir doğal su birikintisinin yanındaki alan kıyı	an area of sand sloping down to the water of a sea or lake
tene yumuşaklık vermek ya da güneş yağmur gibi dış etkilerden korunmak için sürülen güzel kokulu merhem	fine powdery material such as dry earth or pollen that can be blown about in the air
sınırlamak eylemi	a rule or condition that limits freedom
bir sanata bir bilime temel olan yön veren ilke kaide	an occupation requiring special education especially in the liberal arts or sciences

Continued on next page

Table A.1 – *Continued from previous page*

Turkish Definition	**English WordNet Definition**
il ilçe gibi yerleşim bölgelerinde iki yanında evler olan caddeye oranla daha dar ya da kısa olabilen yol	a deep and relatively narrow body of water as in a river or a harbor or a strait linking two larger bodies that allows the best passage for vessels
belli bir saatte belli bir yerde iki ya da daha çok kişi arasında kararlaştırılan buluşma	a time period usually extending from friday night through sunday more loosely defined as any period of successive days including one and only one sunday
araştırılıp öğrenilmesi düşünülüp çözümlenmesi bir sonuca bağlanması gereken durum mesele problem	a question raised for consideration or solution
alışılmış olandan umulandan ya da gerekenden eksik niceliği küçük "çok" karşıtı	the quality of having an inferior or less favorable position
herhangi bir iş bir görev için kendini ileri sürme ya da başkaları tarafından ileri sürülme namzetlik	the real physical matter of which a person or thing consists
birini telefonla aramak ve bir şey söylemek	a seat for one person with a support for the back
denemek eylemi sınama tecrübe	the act of rejecting something
patlıcangillerden yaprakları tüylü çiçekleri salkım durumunda vitamince zengin kırmızı ürünü için yetiştirilen bir bitki lycopersicon esculentum	mildly acid red or yellow pulpy fruit eaten as a vegetable
bir görevi bir işi yasaların verdiği olanaklara göre belli koşullarla yürütmeyi sağlayan hak salahiyet mezuniyet	financial aid provided to a student on the basis of academic merit
bir konu ile ilgili bilgi vermek ve bu bilgiler üzerinde tartışmak amacıyla birkaç yetkilinin yönetimi altında düzenlenen toplantı	a prearranged meeting for consultation or exchange of information or discussion especially one with a formal agenda

Continued on next page

Table A.1 – *Continued from previous page*

Turkish Definition	**English WordNet Definition**
kendi isteğiyle görevden ayrılma çekilme işinden ayrılma	withdrawal from your position or occupation
merkezde bulunan ve bir eksenin çevresinde dönebilir kurs ya da çember teker	an urban area with a fixed boundary that is smaller than a city
bireyle ilgili olan bireye özgü olan ferdi	a person who is a member of a partnership
yumuşakçalardan bahçelerde yaşayan sarmal kabuklu küçük hayvan helix	elongated crescent-shaped yellow fruit with soft sweet flesh
kadın ya da erkek çocuğun en ince sesi	the highest female voice the voice of a boy before puberty
omuz başının altında kolun gövde ile birleştiği yer	the part of the body between the neck and the upper arm
üzerinde sıcak ve soğuk su muslukları bulunan porselen sac emaye gibi maddelerden yapılan el yüz bulaşık yıkamaya yarar yer	metal or earthenware cooking vessel that is usually round and deep often has a handle and lid
adaletle iş gören adaletten haktan ayrılmayan hakkı yerine getiren adaletli	a rule or condition that limits freedom
tanıtma filmi	gathering of producers to promote business
bir makinenin herhangi bir taşıtın hızını kesmeye ya da onu durdurmaya yarayan düzenek	a restraint used to slow or stop a vehicle
ara uzaklık	a large distance
kap kılıf sarma	small thin inflatable rubber bag with narrow neck
herhangi birinden	the effect of one thing or person on another
sinema ya da müzikhol sanatçısı yıldız	a three-dimensional work of plastic art

Continued on next page

Table A.1 – *Continued from previous page*

Turkish Definition	**English WordNet Definition**
okuyup yazmadan başlayarak en yüksek düzeyde bilim ve sanat bilgisi vermeye değin çeşitli derecede toplu olarak öğrenimin sağlandığı yer mektep	an occupation requiring special education especially in the liberal arts or sciences
başın altına koymak ya da sırtı dayamak için kullanılan içi yün pamuk kuştüyü gibi şeylerle doldurulmuş küçük minder	a piece of cloth used as decoration or to mend or cover a hole
alt ve üst tabanları birbirine eşit dairelerden oluşan bir nesnenin eksenini dikey olarak kesen birbirine koşut iki yüzeyin sınırladığı cisim üstüvane	a support that consists of a horizontal surface for holding objects
bir kimsenin belli bir sürede ya da yaşam boyu edindiği bilgilerin tümü tecrübe	the real physical matter of which a person or thing consists
yaşantıları öğrenilen konuları bunların geçmişle ilişkisini bilinçli olarak saklama gücü hafıza	a storage device on which information sounds or images have been recorded
geçmeye engel olacak biçimde uzunlamasına kazılmış derin çukur	the general feeling that some desire will be fulfilled
bıçak makas gibi bir araçla bir şeyi ikiye ayırmak	a top as for a bottle
yitme yitim	a feeling of restless agitation
eski çağlardan beri söylenegelen olağanüstü varlıkları olayları konu edinen imgesel öykü söylence	the series of events that form a plot