Budapesti Mûszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Department of Automation and Applied Informatics

Universal embeddings

Diplomaterv

Készítette
Eszter Iklódi

Konzulens
Gábor Recski

April 10, 2018

Contents

Kivonat **4**

Abstract **5**

1 Introduction **6**
1.1 Natural Language Processing . . . 6
1.1.1 Common tasks of NLP . . . 6
1.1.2 Motivation for NLP research . . . 7
1.2 Universal Embeddings . . . 8

2 Word embeddings **9**
2.1 Semantic encoding of words . . . 9
2.2 State-of-the-art models for learning word embeddings . . . 10
2.3 Multilingual word embeddings . . . 11
2.3.1 Motivation . . . 12
2.3.2 Tasks . . . 12
2.3.3 Under-resourced languages . . . 15
2.3.4 State-of-the-art models . . . 15
2.4 Multilingual data . . . 18
2.4.1 English-Italian setup of Dinu . . . 19
2.4.2 The `fastText` embedding . . . 20
2.4.3 Panlex . . . 20

3 Proposed model **25**
3.1 Description of our method . . . 25
3.1.1 Cosine similarity and precision . . . 25
3.1.2 Equation to optimize . . . 26
3.1.3 Configuration parameters . . . 26
3.2 Software architecture . . . 28

4 Experiments **30**
4.1 Summary of the experiments . . . 30
4.1.1 Dataset creation . . . 30
4.2 Baseline experimental setting . . . 30

4.2.1 Adjusting parameters . . . 30
4.3 Description of different tasks . . . 30
4.3.1 Word translation . . . 30
4.3.2 Cross-lingual semantic word similarity . . . 30
4.4 Summarizing the contributions of the thesis . . . 31

5 Future work **32**

Köszönetnyilvánítás **33**

Irodalomjegyzék **37**

HALLGATÓI NYILATKOZAT

Alulírott *Eszter Iklódi*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot/ diplomatervet (nem kívánt törlendõ) meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelmûen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerzõ(k), cím, angol és magyar nyelvû tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhetõ elektronikus formában, a munka teljes szövegét pedig az egyetem belsõ hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetõvé.

Budapest, April 10, 2018

Eszter Iklódi
hallgató

Kivonat

Mindennapi életünkben egyre fontosabb szerepet tölt be a természetes nyelv számítógép segítségével történő feldolgozása. Digtitalizált világunkban egyre inkább alapkövetelmény, hogy a gép és ember közötti kommunikáció természetes nyelven történjen. Ennek a megvalósításához elengedhetetlen az emberi nyelv szemantikai értelmezése.

Manapság a state-of-the-art rendszerekben a szavak szemantikai reprezentációja sokdimenziós vektorokkal, word embedingek-kel történik. Diplomaterv munkámban már feltanított word embeddingek-hez keresek olyan fordítási mátrixokat, amelyek képesek egy adott nyelvű word embedding univerzális térbe történő leképzésére.

Az így nyert fordítási mátrixokat különböző feladatokon értékelem ki. (eredmények 2 mondatban)

Abstract

Computer-driven natural language processing plays an increasingly important role in our everyday life. In our digital world using natural language for human-machine communication has become a basic requirement. In order to meet this requirement it is inevitable to analyze human languages semantically.

Nowadays state-of-the-art systems represent word meaning with high dimensional vectors, i.e. with word embeddings. In my thesis work I am searching for translation matrices to pre-trained word embeddings, such that the translation matrices will be able to map these embeddings into a universal space.

The obtained translation matrices will be evaluated on different tasks. (2 sentences about results)

Chapter 1

Introduction

1.1 Natural Language Processing

In this section I summarize the main motivations and tasks of the field of Natural Language Processing based upon Dan Jurafsky's *Speech and language processing* book [27], and the *Natural Language Processing with Deep Learning* course held by Christopher Manning and Richard Socher, professors of Stanford University School of Engineering. This course is available online on youtube for free [1].

Natural language processing (NLP) is a vibrant interdisciplinary field with many different names, all reflecting a different facet of it. It is often referred to as speech and language processing, human language technology, computational linguistics, or speech recognition and synthesis. The main goal of this field is to get computers to be able to understand and express themselves in human languages.

Natural Language Processing is hard since it deals with what is considered to be one of the most delicate characteristics of human being: with human language. This field is strongly connected with artificial intelligence since the way humans think and feel about the world is mainly happening in terms of human languages.

Being nowhere near as fast as digital channels, expressing ourselves by means of human languages is a very effective way of communication, though. Saying only the minimum message our listeners can fill up the rest with their world and common knowledge, and can easily figure out the missing or misunderstood parts from the context of the situation. This way they are also able to resolve ambiguities, homonyms etc. without even noticing it. Nonetheless, these tasks for a computer are not at all that trivial.

The importance of computer integrated human language communication has gone as far as assigning truly intelligent machines to the ability of being capable of processing language as skillfully as humans do. This idea was first introduced by Alan Turing (1950) who proposed what has come to be known as the Turing test.

1.1.1 Common tasks of NLP

Natural language processing comprises a wide variety of tasks. Some of them like spam detection, POS-tagging, or named entity recognition are considered to be mostly solved problems. Applications for these tasks are now out in the market and are usually integrated to our smart devices even by default.

With some other tasks great progress has been made recently witch implies the existence of already fair enough applications but means that research work is still has to be done. Among them there are tasks like sentiment analysis, words sense disambiguation, syntactic parsing, machine translation etc.

What is still considered to be very hard is to understand the meaning of a text. There are numerous interesting tasks for example question answering, dialogues, summarization, paraphrases, or text inference, for which in order to make relevant progress dealing with the semantics is inevitable.

1.1.2 Motivation for NLP research

We are living in an era when natural language processing is becoming more and more integrated into our every life. With the advent of smart phones the importance of language has gone even further. These devices are having small and rather inconvenient keyboards thus speech-driven communication seems very appealing. Big companies like Amazon, Apple, Facebook, Google, etc. are all putting out of products that use natural language to communicate with users. Since the contributions of this thesis are aiming the research field of word meaning and universal semantic representations, below I will only list applications that can directly take advantages of these contributions.

Speech-driven assistance applications can really make our everyday life more comfortable and more convenient. Let's just consider the fact how much help they could provide to people living with disabilities. These systems include speech input for which at first automatic speech recognition technologies should be applied. But after that, in order to understand the goal of the user, we must run a semantic analysis as well.

An early version of conversational agents and certain strongly domain-based chatbots are already out on the market, providing 24 hour, immediate assistance for customers. By letting computers do the monotone and not at all creative tasks employees could have more interesting jobs, jobs that only humans can do, or they could have less working hours in a week. Any of these would be a great progress for the society [2].

Advances in machine translation has already created a world where non-English speakers can also enjoy the benefits of the English-based web services. Generally, we can say that for widespread languages machine translation has already reached a fairly usable state, for rare languages, however, it is still facing difficulties.

There are also numerous Web related tasks that are strongly relying on the semantic analysis of the text. One promising task is for example the Web-based question answering task which is basically an extended version of the classical Web search, whereby instead of searching just for key words it would also be possible to ask complete questions and thus communicate with the search engine just like human beings do. For all these applications, however, it is inevitable to look way behind the syntactic surface and dig deep into the underlying semantics.

1.2 Universal Embeddings

Given the need for robust representations for many languages, the question of whether human conceptual structure is universal has recently gained interest not only among cognitive scientists ([41], [30], [24]), but among computational linguists as well. Youn et al. [50] has shown that human conceptual structure is independent of certain non-linguistic factors such as geography, climate, topology or literary traditions. Based on such findings we propose a procedure to construct a universal semantic representation in form of translation matrices that serve to map each language to the universal space. We use the pre-trained fastText word embeddings [17], which are available for 294 languages.

TODO: summary of experiments and results.

Github links, self-references, contributions

The thesis is structured as follows:

- **Chapter 1** briefly explains the goals and motivations of the research field of natural language processing. It also summarizes the main contributions and the results of the accomplished work.
- **Chapter 2** discusses the state-of-the-art semantic word representations, the word embeddings. It briefly presents the standard learning procedure for monolingual word vectors (word2vec) and it introduces the concept of multilingual word-embeddings. It also describes the available resources for multilingual embedding learning that were utilized during this work.
- **Chapter 3** describes the proposed model. It details the learning procedure and discusses the basic infrastructural and architectural features.
- **Chapter 4** presents all the experiments. It summarizes our results and compares them with the performance of other systems.
- **Chapter 5** is denoted to the description of the future work. This chapter suggests modifications, follow-ups for which we either didn't have time to accomplish, or which are beyond the scope of this thesis work.

Chapter 2

Word embeddings

2.1 Semantic encoding of words

Within the field of natural language processing a more specific area concentrates on semantic representations which are being leveraged both by classical semantic tasks such as question answering or chatbots and by other NLP tasks which in the strict sense of the word are not considered semantic tasks such as machine translation or syntactic parsing. A crucial part of all semantic tasks is to have a proper word representation which is capable of encoding the meaning as well.

One way to build a semantic representation is to use a distributional model. The idea is based upon the observation that synonyms or words with similar meanings tend to occur in similar contexts, or as it was articulated by Firth in 1957: *You shall know a word by the company it keeps* [23]. For example in the following two sentences `‘‘The cat is walking in the bedroom’’` and `‘‘A dog was running in a room’’` words like `‘‘dog’’` and `‘‘cat’’` have exactly the same semantic and grammatical roles therefore we could easily imagine the two sentences in the following variations: `‘‘The dog is walking in the bedroom’’` and `‘‘A cat was running in a room’’` [16]. Based upon this intuition, what distributional models are aiming to do is to compute the meaning of a word from the distribution of words around it. [27]. The obtained meaning representations are usually high dimensional vectors, called as word embeddings, which refers to their characteristic feature that they model a world by embedding it into a vector space.

Embeddings are not only proved to be a better alternative of n-grams used for language modelling [16], but they are doing quite well on semantic tasks too. Mikolov et al. [34] has shown that the characteristics of word embeddings go way beyond the simple syntactic regularities. They showed that applying simple vector operations (e.g. vector addition and subtraction) can often produce meaningful results. For example it was shown that *vector("King") - vector("Man") + vector("Woman")* results in a vector that is closest to the vector representation of the word *Queen* [35]. Moreover, these days' state-of-the-art results on work similarity tasks are all held by word embeddings, where the similarity of two words are calculated as the normalized dot product of the corresponding word vectors. This measure is the so-called cosine similarity of word embeddings.

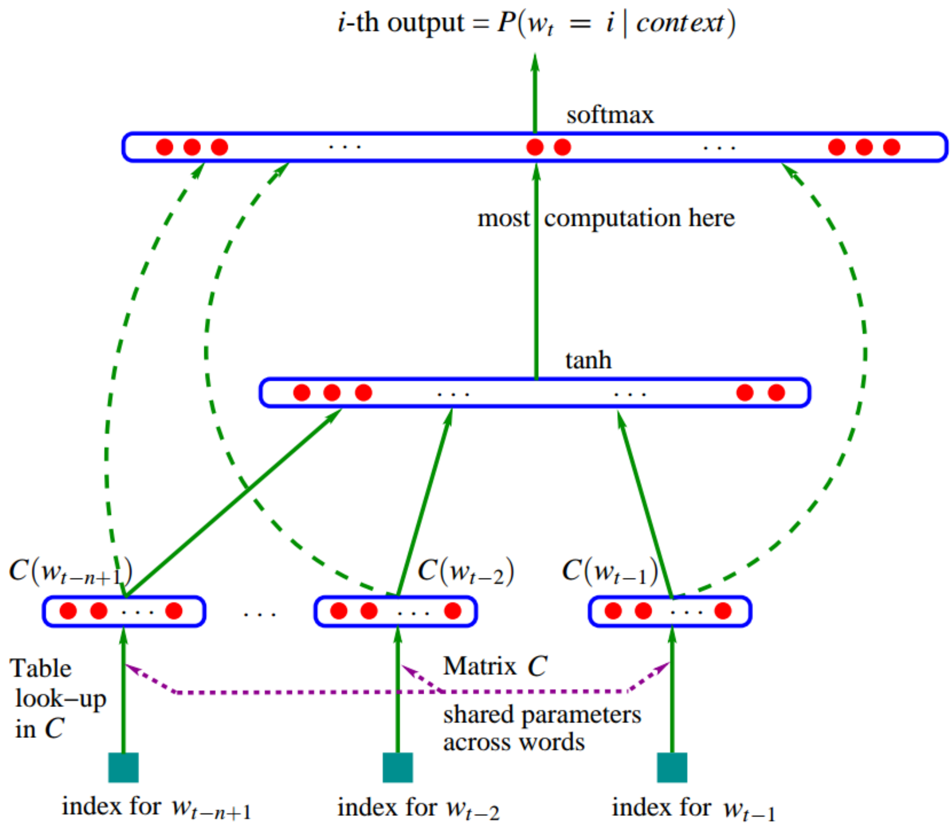
Another way to build semantic representations is to utilize lexical databases. In our previous work we created a hybrid system leveraging both the `4lang` othologycal model described in [28], [29] and [10] and various distributional models, i.e. various word embeddings. With this system in 2016 we reached a state-of-the-art score on the `SimLex-999` [26] benchmark data [40].

In the following sections I will describe the basic procedure of training word embeddings and following that I will focus on a more specific field of computational semantics, on multilingual word embeddings.

2.2 State-of-the-art models for learning word embeddings

In 2003 Bengio et al.[16] suggested a probabilistic feedforward neural network language model (NNLM) for learning a distributed representation for words. The network consists of input, projection, hidden and output layers, where at the input layer the N previous words are encoded using 1-of-V coding, where V is the size of the vocabulary. Figure 2.1 shows an overview of the architecture. This procedure was evaluated in terms of perplexity at which in comparison with the best of the n-grams it proved to be much better. The drawback of this architecture is that it becomes complex for computation between the projection and the hidden layer, as values in the projection layer are dense.

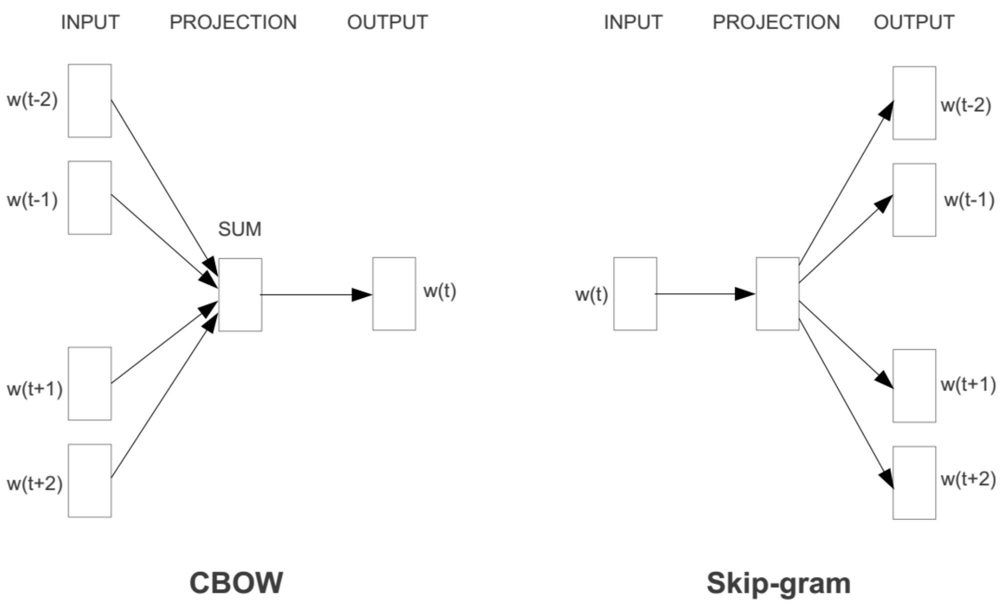
Figure 2.1: *Network architecture proposed by Bengio et al.[16]*



10 years later, in 2013 Mikolov suggested a bag-of-words neural networks, more specif-

ically two different architecture [32]. The first one, denoted as the CBOW (Continuous Bag-of-Words Model) tries to predict the current word based on the context, whereas the second one, denoted as the continuous skip-gram model tries to maximize the classification of a word based on another word in the same sentence. Both models worked better than the NNLM suggested by Bengio [16] both on semantic and syntactic tasks, while between the two models of Mikolov the CBOW turned out to be slightly better on syntactic tasks and the skip-gram on semantic tasks. Mikolov's procedure has become known as the `word2vec` procedure and the source code is available on github http://deeplearning4j.org/word2vec. The architecture of the CBOW and the skip-gram models are shown in figure 2.2.

Figure 2.2: *Bag-of-words neural networks suggested by Mikolov et al.[34]*



Embeddings are usually evaluated on word similarity and word analogy tasks. Besides providing quite promising results on them, they have also been applied to many downstream tasks, from **named entity recognition** and **chunking** [48] to **dependency parsing** [14]. It has furthermore been shown that weakly supervised embedding algorithms can also lead to huge improvements for tasks like **sentiment analysis** [46].

2.3 Multilingual word embeddings

In this section I describe the importance of multilingual word embeddings. I also explain how it is possible to incorporate word embeddings trained on monolingual text corpora into a multilingual context. After that I present a brief summary about the previous attempts on constructing cross-lingual word vector representations. Finally, I describe the available parallel datasets that we were using for our experiments.

2.3.1 Motivation

The question how to model representations is a highly interdisciplinary issue to discuss. Within cognitive science, traditionally there are two dominating approaches to this problem. The first one is the *symbolic* which states that cognitive systems can be described as Turing machines. The second one, denoted as *associationism*, says that representations are associations among different kinds of information elements. In his book, *Conceptual Spaces: The Geometry of Thought* [24],Gärdenfors advocates a third approach, which he calls *conceptual* from. This representation is based on using geometrical structures rather than symbols or connections among neurons.

To go a step further one could ask whether these structures are universal among all human beings. Regarding this question with the eyes of a computer scientist we might form this problem as whether it is possible to model meaning universally, i.e. language independently. Current meaning representations are leaned from monolingual corpora, therefore they infer language dependency. But is there a way to find one single representation instead of a different one for each and every human language?

Youn et al. [50] suggested that human brain may reflect distinct features of cultural, historical, and environmental background in addition to properties universal to human cognition. They provided an empirical measure of semantic proximity between concepts using entries of the Swadesh list [45]. The Swadesh list is cross-linguistic dictionary which includes a 110 and a 207 long list of basic concepts in approximately 2000 languages. Youn et al. took 22 concepts of this list that refer to material entities (e.g. STONE, EARTH, SAND, ASHES), celestial objects (e.g., SUN, MOON, STAR), natural settings (e.g., DAY, NIGHT), and geographic features (e.g., LAKE. MOUNTAIN). Then, they applied translation and back-translation through various languages. As a result of numbers of polysemies in the resulting graph originally distinct concepts become connected. For example the Spanish word *cielo* in English both means *heaven* and *sky*. Thus by applying English-Spanish-English translation and back-translation the two English words *heaven* and *sky* become connected. The more such polysemous words we find, the stronger this connection becomes. For example if besides Spanish, we also apply the translation and back-translation through German, the same polysemy appears: the German word *Himmel* both means *heaven* and sky in English. The procedure is shown on figures 2.3 and 2.4.

Statistical analysis of the obtained graphs constructed over the polysemies observed in the above-mentioned 22-word-long subset of basic vocabulary showed that the structural properties of these graphs are consistent across different language groups, and largely independent of geography, environment, and the presence or absence of a literary tradition. Based upon these findings we assume that meaning, at least to a certain extent, is universal, thus representing semantics at universal level is reasonable.

2.3.2 Tasks

Beyond the theoretical level of whether meaning is universal there are numerous practical problems for which cross-lingual embeddings might come in handy. In this section I

Figure 2.3: *Translating MOON and SUN through polysemous words.*

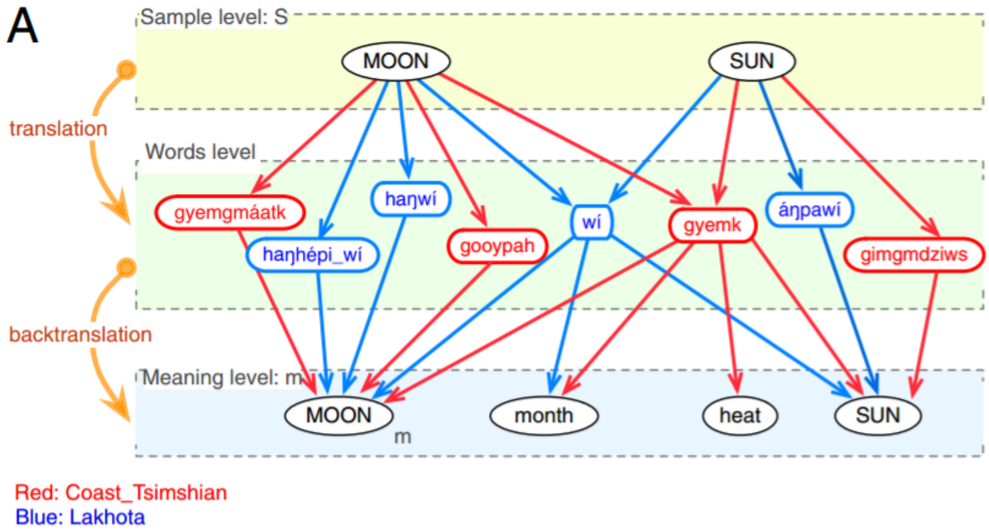
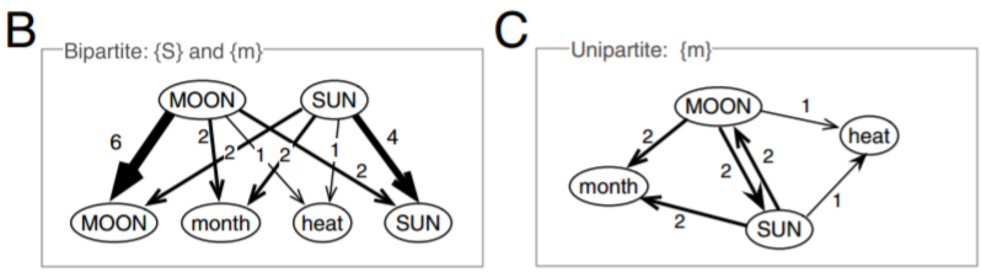


Figure 2.4: *Making links between English concepts through eliminating the internal nodes.*



write about the different tasks where solutions can be facilitated by utilizing multi-lingual embeddings.

Cross-language part-of-speech tagging

Part-of-speech tagging, a.k.a. POS-tagging is the task for annotating a text with part-of-speech tags. The fundamental idea behind the multilingual learning of part-of-speech tagging is that when assigning part-of-speech tags the patterns of ambiguity differ across languages. A word with part-of-speech tag ambiguity in one language may correspond to an unambiguous word in the other language. For example, the word "can" in English may function as an auxiliary verb, a noun, or a regular verb, however, translating the sentence into other languages the different meanings of "can" are likely to be expressed with different lexemes. By combining natural cues from multiple languages, the structure of each POS-tagger becomes more apparent [37].

Todo: ref to applications of multiling embeddings, is there any ??

Cross-language super sense tagging

The Semantic Web paradigm is often required to provide a structured view of the unstructured information expressed in texts. In order to cover the web scale in almost any languages, abundance of such knowledge is required. More specific fields of Natural Language Processing such as ontology learning and information extraction are focusing on finding solutions to this difficult problem.

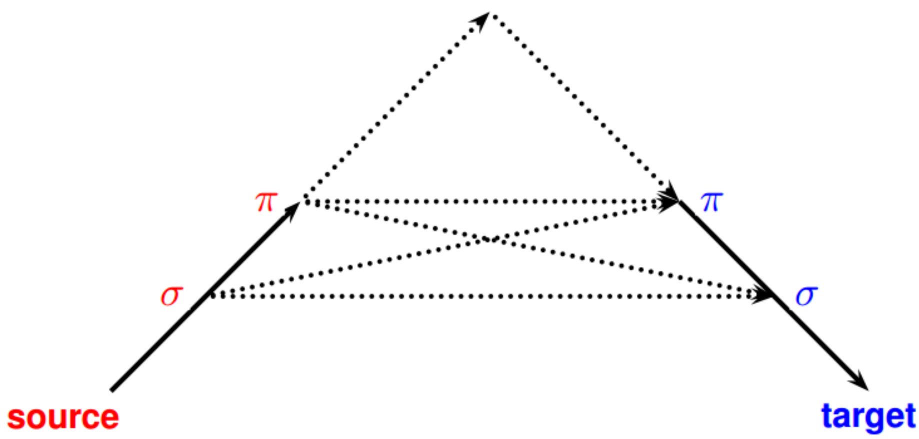
SuperSense Tagging is the problem of assigning "supersense" categories (e.g. `person`, `act`) to the senses of words according to their context in large scale texts. Opposite to NER (Named Entity Recognition) systems a Super Sense Tagger does not make a difference between proper and common names. These "supersense" categories include 41 general concepts defined by WordNet [15], which originally introduces 45 lexicographer's categories [21]. The idea behind WordNet is to provide lexicographers an initial broad classification for lexicon entries.

Attempts for creating such systems have already been made. For example Picca et al. [38] trained a multilingual super sense tagger on the Italian and English languages. Despite the fact that they did not use any word embedding, the introduction of multilingual word embeddings to this task could significantly facilitate the development of multilingual knowledge induction, ontology engineering, and knowledge retrieval.

Machine translation

Machine translation is the task of translating a text automatically with a computer from a source language to a target language. Current research works are focusing on finding the appropriate level of representations when performing the translations. Basic approaches are: `tree-to-string`, `string-to-string`, and `string-to-tree`, as shown in figure 2.5.

Figure 2.5: *Levels of representation in Machine Translation.* $\pi \rightarrow \sigma : tree-to-string; \sigma \rightarrow \sigma : string-to-string; \sigma \rightarrow \pi : string-to-tree.$



Translation models, however, often fail to generate good translations for infrequent words or phrases. Previous works attacked this problem by inducing new translation rules from monolingual data with a semi-supervised algorithm. Nevertheless, this approach does not scale very well since computationally it is quite expensive. Zhao et al. [51] proposed a much faster and simpler method that creates translation rules for infrequent phrases based on phrases with similar continuous representations, i.e. with similar word vectors, for which a translation is known. Their method improved a phrase-based baseline by up to 1.6 BLEU on Arabic-English translation, and it was three-orders of magnitudes faster than existing semi-supervised methods and 0.5 BLEU more accurate.

By introducing a universal vector space, in order to cover all possible translation pairs for n languages, instead of having to train $\binom{n}{2}$ translators it would be enough to train only $2n$ translators, for each language from the source space to the universal space and vica versa, which would significantly simplify the Machine Translation task.

2.3.3 Under-resourced languages

Dictionaries and phrase tables are the basis of modern statistical machine translation systems. Mikolov et al. [33] showed a method that can automate the process of generating and extending dictionaries and phrase tables. They could translate missing word and phrase entries by learning language structures based on large monolingual data and mapping between languages from small bilingual data. This is a powerful opportunity for rare languages to join the mostly English-based world of the Web and for non-English speakers to enjoy its benefits without having to speak English.

2.3.4 State-of-the-art models

In this section I present a brief history on cross-lingual word vector representations. As a baseline approach I describe the procedure of Mikolov et al. [33] from 2013 and next I study various attempts made since then to improve this baseline system and to alleviate its errors. Finally, I summarize some recent attempts for obtaining multilingual word embeddings without using any parallel data.

First attempt: Mikolov et al.

Right after publishing their `word2vec` procedure, Mikolov et al. [33] went even further by noticing that continuous word embedding spaces exhibit similar structures across languages. They applied a simple two-step procedure:

- firstly, monolingual models of languages using huge corpora are built, e.g. by using the `word2vec` method
- secondly, a small bilingual dictionary was used to learn linear projection between the languages. These words are often referred to as anchor points.
- finally, at test time, the translation of any word from the source language is possible by projecting its vector representation from the source language space to the target language space. Once the vector in the target language space is obtained, the most similar word vector can serve as the output of the translation.

With applying only the translation matrices they achieved **51% precision@5** for translation of words between **English and Spanish**. To obtain dictionaries first they created monolingual corpora from the **WMT11** text data [8]. Then they took the most frequent words from these monolingual source datasets, and translated them using on-line **Google**

Translate (GT). Beside simple words, they also used short phrases as the dictionary entries.

In addition to the promising result on the English-Spanish word translation task, this method seemed to be working even for distant language pairs like English and Vietnamese.

In this work for the first step I used a pretrained word embedding described in 2.4.2. The work itself concentrates on finding the linear projections and on the evaluation of these projections.

Various improvements

Faruqui and Dyer

Since Mikolov's experiments various attempts were made to improve the cross-lingual embeddings. Faruqui and Dyer [20] intended to gain information from the translation of a given word in other languages. The most obvious solution would be to append the two word vectors coming from the two languages, but this procedure is highly exposed to drawbacks such as increases in dimension, introduction of irrelevant data, and incapacity of generalization across languages. To counter these problems they used **canonical correlation analysis (CCA)** which is a way of measuring the linear relationship between two multidimensional variables. It finds two projection vectors, one for each variable, that are optimal with respect to correlations, with preserving or even reducing the dimansionality. This way they obtained multi-lingual embeddings based on monolingual embeddings, which they tested on four different standard **word similarity tasks**:

- On the **WS-353** dataset [22] that contains 353 pairs of English words that have been assigned similarity ratings by humans. This dataset was later further divided into two different fragments *similarity*, **WS-SIM**, and *relatedness*, **WS-REL** by Agierre et al. [11] who claimed that these two different kinds of relations and should be dealt with separately
- On the **RG-65** dataset which contains 65 pairs of nouns ranked by humans [42].
- On the **MC-30** dataset which contains 30 pairs of nouns ranked by humans [36].
- On the **MTurk-287** dataset [39] that constitutes of 287 pairs of words that has been constructed by crowdsourcing the human similarity ratings using Amazon Mechanical Turk.

These word representations obtained after using multilingual evidence performed significantly better on the above mentioned evaluation tasks compared to the monolingual vectors. The method was more suitable for semantic encoding than for syntactic encoding. As a conclusion, they showed that multilingual evidence is an important resource even for purely monolingual applications.

Xing et al.

Xing et al. [49] has shown that bilingual translation can be largely improved by **normal-**

izing the embeddings and by restricting the transformation matrices into **orthogonal** ones.

For a comparison, they largely followed Mikolov's settings [33] for creating an English-Spanish dictionary. After extracting the monolingual datasets from the **WMT11** corpus they selected the 6000 most frequent words in English and employed the online Google's translation service to translate them to Spanish. The resulting **6000 English-Spanish** word pairs are used to train and test the bilingual transform in the way of **cross validation**. First, they reproduced Mikolov's results and then they showed that their method outperforms those results with approximately 10 % on this English-Spanish setting.

Dinu et al.

Dinu et al. [19] studied the phenomenon of **hubs**. He showed that the neighbourhoods of the mapped vectors are strongly polluted by hubs. These vectors tend to be near a high proportion of items, and thus their correct labels will be pushed down in the neighbour list when looking up for word translations. They proposed a simple method to alleviate this problem with which they achieved consistent improvements. Furthermore, they observed that many translations were in fact correct but were not present in the gold dictionary.

The experiments were carried out on an **English-Italian** dataset created by themselves and discussed in detail in 2.4.1.

Lazaridou et al.

Lazaridou et al. [31] studied some theoretical and empirical properties of general cross-space mapping function, and tested them on cross-linguistic (word translation) and cross-modal (image labelling) tasks. By introducing **negative samples** during the learning process they could reach state-of-the-art results on the English-Italian word translation task described in 2.4.1. Settings for the negatives examples were studied both by choosing them random and by choosing "intruders" which are near the mapped vector, but far from the actual gold target space vector. The "intruder" approach achieved better results, and furthermore, it gave better results after just few training epochs. The world translation experiments were carried out on Dinu's **English-Italian** setting, and at that time they reached state-of-the-art results on it.

Ammar et al.

Ammar et al. [12] proposed methods for estimating and evaluating embeddings of words in **more than fifty languages** in a **single shared embedding space**. The so-called multiCluster and multiCCA methods were tested on 59 languages, while the multiSkip and translation-invariance methods only on 12 languages for which high-quality parallel data are available. For the 12 languages the bilingual dictionaries were extracted from the **Europarl** parallel corpora, while for the remaining 47 languages, dictionaries were formed by translating the **20k most common words in the English monolingual corpus with Google Translate.**

Artetxe et al.
Artetxe et al. [13] built a **generic framework** that generalizes previous works made on cross-linguistic embeddings. For evaluating the methods they used the same **English-Italian** dataset by Dinu, discussed in 2.4.1. As a conclusion they published that from the proposed methods the ones with orthogonality constraint and a global preprocessing with length normalization and dimension-wise mean centering achieved the best overall results.

Smith et al.
Smith et al. [44] also proves that translation matrices should be **orthogonal**. They apply **singular value decomposition (SVD)** to achieve this. Besides, they introduce a novel **"inverted softmax"** method for identifying translation pairs, with which they improve the precision of Mikolov. Orthogonal transformations also turned out to be more robust to noise which makes it possible to learn the transformation without expert bilingual resource by constructing a "pseudo-dictionary" from the identical character strings. For evalutaion they also used Dinu's **English-Italian** setting. In order to compare their method with the previous ones they reproduced their experiments both in English-Italian and Italian-English directions, and published a summary in form of a table that I present here as well, as Table 2.7 and Table 2.8. Their results achieved **state-of-the-art** scores on Dinu's dataset.

Without parallel data

While all the above mentioned methods rely on biligual word lexicons, most recent studies are aiming to eliminate the need for any parallel data at all. Smith et al. [44] has already made attempts for the alleviation of parallel data supervision by introducing **character-level information**, but the results were not on par with their supervised counterparts, on the one hand, and on the other hand, these methods are strictly limited to pairs of languages sharing a common alphabet.

Conneau et al. [18] introduces an **unsupervised** way for aligning monolingual word embedding spaces between two languages **without using any parallel corpora**. Their experiments show that this method can be applied even for distant language pairs like English-Russian or English-Chinese.

On Dinu's standard word translation retrieval benchmark, using 200k vocabularies, their method reached **66.2%** accuracy on **English-Italian** while the best supervised approach is at **63.7%**.

2.4 Multilingual data

In this section I briefly describe the data resources that I used. These involve the pretrained embeddings I took for the experiments and the gold bilingual dictionaries I used for the evaluation.

2.4.1 English-Italian setup of Dinu

Dinu et al. [19] has constructed an English-Italian dictionary split into a train and a test that are now being used as a benchmark data for evaluating word translation tasks.

Both train and test translation pairs are extracted from a dictionary built from Europarl, available at http://opus.lingfil.uu.se/ (Europarl, en-it) [47]. For the test set they used 1,500 English words split into 5 frequency bins, 300 randomly chosen in each bin. The bins are defined in terms of rank in the frequency-sorted lexicon: [1-5K], [5K-20K], [20K-50K], [50K-100K] and [100K-200K].

For the training translation pairs they also sampled by frequency, using the top 1K, 5K, 10K and 20K most frequent translation pairs from the Europarl dictionary sorted by English frequency. There is no overlap with test elements on the English side, however, when checking for overlaps with the 5K train data on the Italian side we found 113 Italian words that can be found both in train and test sets. These overlaps can be sorted into the following categories:

- **Singular-plural correspondence:** in Italian when the last vowel of a substantive is accented the plural form is the same as the singular. For example *comunità* and *attività* in table 2.1.
- **Italian word is mistaken for English word:** the English translation is the same as the original Italian word. For example in the test set the Italian word *segni* is not translated and the same happens with *vecchi*. See table 2.2.
- **Different verb forms:** the same Italian word can be translated into different English verb tenses. For example *sostenere* in table 2.3.
- **Synonyms and homonyms:** one Italian word can be translated into several English words that might be synonyms or might not in case of homonyms. This phenomenon is actually fairly understandable and acceptable under all circumstances. See table 2.4.
- **Errors in the translation:** wrong translations. For example plural form Italian words *gatti* and *passeggeri* are translated both as the correct plural form and the incorrect singular form. See examples in table 2.5.

Table 2.1: *Singular-plural correspondence*

Italian	**English - train**	**English - test**
comunità	communities	community
attività	activities	activity

A summary of word counts can be seen in table 2.6.

Smith et al. [44] reported results on this English-Italian dataset both in English-Itanlian and Italian-English direction. They reproduced the methods of Mikolov [33], Faruqui [20] and Dinu . A summary of the English-Italian results can be found in table 2.7 and the

Table 2.2: *Italian word is mistaken for English word*

Italian	**English - train**	**English - test**
segni	signs	**segni**
vecchi	old	**vecchi**

Table 2.3: *Different verb forms:*

Italian	**English - train**	**English - test**
sostenere	support	supporting

Italian-English in table 2.8, respectively. All the methods turned to be more accurate when translating from English to Italian. This is not surprising at all, given the fact that many English words can be translated to either the male or female form of the Italian word.

2.4.2 The `fastText` embedding

Usual techniques for obtaining continuous word representation, i.e. word embeddings, represent each word of the vocabulary by a distinct vector, without parameter sharing. They ignore completely the morphology of words which is a significant limitation especially for agglutinating languages, e.g. Hungarian. In these languages new words are formed by stringing together morphemes which leads to large vocabularies and many rare words.

In 2017 the Facebook AI Research group proposed a new approach based on the skipgram model [32], but this time, contrary to the previously mentioned methods, parameter sharing was applied, since words are represented as a bag of character n-grams [17]. A vector representation is associated to each character n-gram, and the words are being represented as the sum of these representations. The method turned out to be fast and it allows us to compute word representations for words that did not appear in the training data. The model was evaluated both on word similarity and analogy tasks. The results show that this model outperforms Mikolov's `CBOW` and `skipgram` baseline systems that do not take into account subword information. It also does better than methods relying on morphological analysis.

The pre-trained word vectors for 294 languages, trained on Wikipedia using fastText are available on the following github link:
https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md.

2.4.3 Panlex

"Overcoming Language Barriers by Connecting Every Word in Every Language" that is the motto of PanLex, which is a nonprofit organiization with a mission to overcome language barriers to human rights, information, and opportunities [6]. They have been building a lexical database for over 10 years now, by transforming thousands of dictionaries into a single common structure. Thus the PanLex database is a very powerful resource that I use in this work as gold data.

The name PanLex is coming from of the words *panlingual* and *lexical.* The former one

Table 2.4: *Synonyms and homonyms*

Italian	English - train	English - test
risposte	answers	responses
sufficiente	sufficient	enough

Table 2.5: *Errors in the translation*

Italian	English - train	English - test	Explanation
gatti	cat	cats	it only means cats
passeggeri	passengers	passenger	it only means passengers

Table 2.6: *Statistics of word counts. The train set contains 5000 and the test set 1500 word pairs, respectively.*

Language	Set	# words
eng	train	3442
	test	1500
ita	train	4549
	test	1849

Table 2.7: *English to Italian results on Dinu's data published by Smith*

Precision	@1	@5	@10
Mikolov et al. (2013b)	0.338	0.483	0.539
Faruqui et al. (2014)	0.361	0.527	0.581
Dinu et al. (2015)	0.385	0.564	0.639
Smith et al. (2017)	**0.431**	**0.607**	**0.664**

Table 2.8: *Italian to English results on Dinu's data published by Smith*

Precision	@1	@5	@10
Mikolov et al. (2013b)	0.249	0.410	0.474
Faruqui et al. (2014)	0.310	0.499	0.570
Dinu et al. (2015)	0.246	0.454	0.541
Smith et al. (2017)	**0.380**	**0.585**	**0.636**

emphasizes the coverage of every single language, while the second one refers to its aim of translating words instead of sentences. Google Translate and other machine translation applications translate whole sentences and texts in up to a hundred major world languages, whereas PanLex focuses only on words translation, but in thousands of languages. The PanLex database is free for noncommercial use, and the translator itself can also be found online on the following website: https://apps.panlex.org/translator/.

The following link, https://panlex.org/source-list/, lists all the dictionary sources that are registered to the PanLex database. All of them have been acquired which means that either files have been downloaded, or books were purchased, scanned and adapted, but not all have been analyzed and incorporated into the PanLex database yet. It is possible to query specific sources, and for every entry we find the author, the publish year and the number of entries of a specific dictionary resource.

From a developer's view the PanLex project is focusing on the following tasks: *acquisi-*

tion, assimilation, interface creation, research, infrastructure development. Below I briefly summarize these above mentioned tasks, and after that I emphasize the most important features of the data model and the database design of PanLex. To participate in the project you can apply as an itern or as a volunteer.

Acquisition Acquisition is the work of acquiring data for PanLex. The most important concept is the *expression* which refers to words and word-like phrases. Words-like phrases obviously contain more words, but the reason for that is that these phrases are necessarily have to be treated as whole, since word-by-word translation does not apply. What we can still consider as a word-like phrase is not quite clear so far, for example, one can debate whether "orange juice" counts as an *expression*, but "sweet potato" surely does. As for languages with various inflections (e.g. number, case etc.) it must be known that *expressions* only appear in their "lexeme" form. Translations can be unilingual (synonyms), bilingual, or multilingual, from which, keeping in mind the main purpose of the PanLex project, multilingual dictionaries are the most valuable. Furthermore, if an *expression* has translations into 20 different languages, then, than instead of only 20 translations, in fact 210 ones are found.

The PanLex database has over a billion lexical translations if counting each pair of expressions between which there is at least one attested translation. Nonetheless, there is great disparity among languages. There are about 2,000 languages (recognized by the ISO 639-3 standard, http://www-01.sil.org/iso639-3/) with no expressions at all, and most of the "threatened" or "endangered" languages are reflecting very "low-density". The priorities of acquisition are as follows:

- coverage: the more languages the better
- diversity: poorly documented language families have high priority
- laterality: still not contained dictionary pairs are more useful
- quality: qualified resources tend to be more useful
- efficiency: easy-to-convert data is preferred

Own translations are not accepted, they aim to collect already existing translations made by domain experts. The ownership of data is slightly controversial, but protection from public access and fair use of copyrighted works is guaranteed.

To facilitate the effort of the acquisition process various tools are provided. As for the project management system they use Wrike [9].

Assimilation Assimilation is the process of consulting the acquired data and to convert it to a form that complies with the standards of the PanLex database. The method of assimilation can vary greatly from one source to another, which might include using different types of programming languages, the application different keyboards (e.g. Arabic, Chinese), and sometimes it is inevitable to understand the language itself in order to assimilate is correctly.

Interfaces The PanLex data can be accessed through various interfaces. Although interface development for human users is only a secondary purpose of PanLex, (since the primary one is to enrich the lexical database), there are still quite a few different available interfaces, including an online translator, standard HTTP API, custom search engines, and an integration with the NLTK [5] python library, called PanLex lite.

In addition, there are also some games that are being developed to make the exploration of PanLex expressions more interesting and to invite more people to use PanLex.

Research Current research fields, like translation inference, cross-language information retrieval, or cross-language human communication can make use of PanLex data. The PanLex itself also began as an applied research project. Current research toping regarding PanLex are focusing on the following areas:

- quality management: supervision of source data
- error detection: automated detection of errors in source data
- translation confidence: improvement of translation estimates that PanLex assigns to translations
- ontological enrichment: extension with language varieties, addition of conceptions, e.g. Morris Swadesh' "Swadesh list" [45]
- visualization: automated visualization of the PanLex database

Infrastructure Behind the database itself there is an extended infrastructure as well. Developments are carried out to improve the capabilities, design, performance and reliability.

Data model A detailed description of PanLex data model is available at https://dev.panlex.org/data-model/.
Here I would only like to explain the concept of a *language variety*. A *language variety* consists of a *language* and a three-digit *variety code*.

- *language*: is a three-letter ("alpha-3") code which complies with the ISO 639 protocol. For example in case of the English language the code is **eng** or as for Hungarian it is **hun** etc.
- *variety code*: is a three-digit code starting with **000**.
- *language variety = "language"-"variety code"* e.g. **eng-000** designates English. The most widely spoken variety of a language is often variety **000**, but there is no systematic rule for the assignment of variety codes. Variety codes include (but are not limited to) regional variation and different writing systems. For example, **eng-004** designates American English.

In our experiments we always take the *language variety* with the smallest *variety code*.

Database design A detailed description of the PanLex database design is available at https://dev.panlex.org/database-design/.

What we need to extract from the PanLex database are expression pairs between two languages. For that we need to read the `langvar` table (language varieties, abbreviated as **lv**) and the `expr` (expressions abbreviated as **ex**) table.

The code for extracting the expression pairs and creating a tsv file from them is available at:

https://github.com/Eszti/dipterv/blob/master/panlex/scripts/panlex/extract_tsv.py

TODO: what does score mean??

Chapter 3

Proposed model

3.1 Description of our method

In this section I describe our proposed model in detail. I define the metrics that we optimize during training and evaluation processes, I introduce the equation we use for optimizing and I discuss the different configuration parameters of the system.

In a nutshell, in our work we are searching for linear mappings in form of translation matrices between pre-trained word embeddings through a universal vector space. We divide gold dictionary data into training, development and test subsets. During training we maximize the cosine similarity of word translations in the universal space. After having mapped the embeddings of two different languages into this universal space the cosine similarity of the actual translation pairs should be high, i.e. close to 1. At test time we evaluate our system with the precision metric used principally for word translation tasks.

3.1.1 Cosine similarity and precision

In word similarity tasks the extent to which the meanings of two words are similar is intrigued. **Cosine similarity** [3] is a measure of similarity between two non-zero vectors. It is calculated as the inner product of the two vectors divided by the product of the lengths of the two vectors, as shown in Equation **??**. Therefore it is a space that measures the cosine of the angle of two vectors. In word similarity tasks the similarity of two words represented as word vectors are measured by the cosine similarity of the two vectors. The maximum value, +1, denotes complete similarity, 0 denotes neutrality, and -1 corresponds antonyms.

$$cosine\_similarity = cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} \tag{3.1}$$

In word translation tasks the performance is usually measured by the **precision** metric. At a word translation task there is always a lookup space, which, for example, in our experiments corresponds to the most frequent 200k words of the given language. After translating a word we find the N closes words to the translated vector. The metric **Preci-**

sion @N denotes the percentage of how many times the real translation of the word was found among the N closes words from the lookup space. Usual N values are 1, 5, and 10, respectively.

3.1.2 Equation to optimize

The objective of our method is to learn linear mappings in form of translation matrices that are obtained by maximizing the cosine similarity of gold word translation pairs in a universal space. Therefore for each language one single translation matrix is searched that maps the language from its original vector space to a universal one.

Our objective is to bring close the translation pairs in the shared universal space, therefore, our method is applicable not just for language pairs, but for any number of languages. The main advantage is that by introducing new languages the number of the learned parameters remains constant since instead of learning pair-wise translation matrices for each language only one is learned that maps directly to a shared, universal space.

Let L be a set of languages, and TP a set of translation pairs where each entry is a tuple of two in form of (w_1, w_2) where w_1 is a word in L_1 language and w_2 is a word in L_2 language, and both L_1 and L_2 are in L. Then let's consider the following equation, the Equation **??**:

$$\frac{1}{|TP|} \cdot \sum_{\substack{\forall L_1, L_2 \\ \in L}} \sum_{\substack{\forall (w_1, w_2) \in TP, \\ w_1 \in L_1, w_2 \in L_2}} cos\_sim(w_1 \cdot T_1, w_2 \cdot T_2) \tag{3.2}$$

where T_1 is the translation matrix that maps L_1 to the universal space, and T_2 which maps L_2 to the universal space, respectively. Since we normalize the equation with the number of translation pairs in the TP set, the optimal value of this function is 1.

Note: if w_1 and w_2 values are normalized, as Xing et al. [49] suggested, the *cos_ sim* reduces to the simple dot product of the translated vectors. In our experiments we are always working with normalized vectors.

At test time we evaluate our system with the precision metric, more specifically with Precision @1, @5, and @10. The distance assigned to the word vectors in the look-up space is the *cosine_ similarity*. The bigger this value is, the closer the two vectors are.

3.1.3 Configuration parameters

During the training process there are several configuration parameters that we adjusted using the development set. In this section I only recite them, the process of finding the optimal values and the results of experimenting with different setups are discussed in Chapter 4 in detail.

Generic parameters

The optimization process has several parameters that we can adjust. Below I enlist those ones that we experimented with:

- **optimizer:** the method to find the optimum value of the equation. Most common optimizers are: SGD (Stochastic Gradient Descent), Adagrad, Adadelta, Adam, Adamax [4]
- **epochs:** one epoch is the number of iterations after which we have seen each and every example of the training set exactly one time. The more epoch we do, the more the system has learned.
- **batch size:** the number of examples we consider in one iteration. Originally we differentiate three different types of SGD algorithms based on the batch size [7]:
  - **BGD (Batch Gradient Descent)** refers to the procedure when the batch size is equal to the number of all the training examples, i.e. one iteration is the same as one epoch.
  - **SGD (Stochastic Gradient Descent)** refers to the procedure when the batch size is equal to one, i.e. one epoch consists of as many iteration as the number of examples in the training set. It is often referred to as online learning, or noisy gradient descent.
  - **MBGD (Mini-batch Gradient Descent)** is a compromise between BGD and SGD, namely it means that the size of a batch can vary from 1 to N, where N denotes the number of all the training examples.
- **learning rate:** parameter which adjusts how fast is the learning process.
  - If the learning rate is **high** the learning process is faster, since we are heading to the local minimum with bigger steps. The drawback is that this minimum can easily be missed. Since steps are big, what usually happens is that we keep jumping from one side of the minimum to the other side without ever reaching it, or we are even getting further from it.
  - If the learning rate is **low** the learning process is slower, because the steps taken towards the local minimum are smaller. With a smaller learning rate we tend to get closer to the real minimum point, although sometimes it takes so much that it does not worth waiting for it.
  - The task of adjusting the learning rate parameter is to find the **optimal** payoff between the time needed to run the experiments and the quality of the results that the experiments provide.
- **Batch size - learning rate relation:** Goyal et al.[25] studied the behaviour of different batch size and learning rate combinations, running their experiments on the ImageNet database [43]. As a rule of thumb they determined the following relationship between these two parameters: if an experiment with a base batch size b and a base learning rate η terminates in time t, then if we increase the batch size by a factor of k, i.e. batch size $= b \cdot k$, then in order to keep the execution time at t we should apply $\eta \cdot k$ for the learning rate. In this case, in addition to the same execution

times, the two learning processes are also having rougly the same learning curves, i.e. their loss functions over time are the very similar.

Specific parameters

Besides the generic configuration parameters that have to be adjusted basically at all kind of machine learning or optimization tasks, we have some other, more task specific configuration parameters that we experimented with. These are the following ones:

- **SVD:** as Smith et al. [44] suggested applying SVD (Singular Value Decomposition) to the transformation matrices turns out to be quite useful. Therefore we also introduced a setting option whether to apply SVD or not.
- **SVD mode:** I introduces three different mode for experimenting with SVD. More specifically, they are called 0, 1, and 2. 0 means no SVD at all. 1 means doing SVD only at the very beginning (after the first batch), and 2 means doing an SVD regularly, i.e. on every n-th batch.
- **SVD frequency:** when applying SVD mode 2, this option corresponds to the n, i.e. the frequency how often an SVD will be applied on the translation matrices.
- **Embedding limit:** the number of words occuring in an embedding varies from language to language. In order to be able to evaluate the system for different languages equally we always read only the first n lines of the given word embeddings. This way the lookup space will have the same size for every language.

Parameters for evaluation

At test time we used different metrics for evaluation:

- **Precision:** the most important metric for evaluation is the precision. The system is capable of calculating any number of precisions, although in the end we figured that the most reasonable is to only calculate the values that are widely used by others as well, i.e. Precision @1, @5, and @10.
- **Loss:** at training time we optimize for the cosine similarity on the training set. It is also interesting to see what are the result on the test set.
- **Calculating small singular values:** small singular values of a translation matrix are indicators of dimension reduction of the translated space. Definitely we do not want that to happen, so it is also worth checking out the number of small singular values. The limit below which we consider a singular value as a small value is another configurable parameter.

3.2 Software architecture

In this section I describe briefly the software architecture and the used packages, tools and off-the-self codes I utilized during the implementation of the system.

Used packages (tensorflow . . .)

Artetxe writes about SW arch, take a look at it!

Chapter 4

Experiments

4.1 Summary of the experiments

4.1.1 Dataset creation

4.2 Baseline experimental setting

4.2.1 Adjusting parameters

4.3 Description of different tasks

4.3.1 Word translation

Mikolov: 90 percent precision@5 for translation of words between English and Spanish

4.3.2 Cross-lingual semantic word similarity

Facebook (MUSE): SemEval 2017

4.4 Summarizing the contributions of the thesis

Chapter 5

Future work

Köszönetnyilvánítás

Ez nem kötelezõ, akár törölhetõ is. Ha a szerzõ szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelezõ, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Bibliography

[1] https://www.youtube.com/playlist?list=PL3FW7Lu3i5Jsnh1rnUwq$_{T}cylNr7EkRe$6.

[2] https://www.economist.com/technology-quarterly/2017-05-01/language.

[3] Cosine similarity. https://en.wikipedia.org/wiki/Cosine$_{s}imilarity$.

[4] Keras optimizers. https://keras.io/optimizers/.

[5] Nltk. http://www.nltk.org/.

[6] Panlex. https://panlex.org/.

[7] Stanford: Machine learning course. https://www.coursera.org/learn/machine-learning.

[8] Wmt11. http://www.statmt.org/wmt11/training-monolingual.tgz.

[9] Wrike. https://www.wrike.com/.

[10] Judit Acs, Katalin Pajkossy, and András Kornai. Building basic vocabulary across 40 languages. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 52–58, 2013.

[11] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics, 2009.

[12] Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*, 2016.

[13] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, 2016.

[14] Mohit Bansal, Kevin Gimpel, and Karen Livescu. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 809–815, 2014.

[15] Richard Beckwith, Christiane Fellbaum, Derek Gross, and George A Miller. Wordnet: A lexical database organized on psycholinguistic principles. *Lexical acquisition: Exploiting on-line resources to build a lexicon*, pages 211–232, 1991.

[16] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[17] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[18] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[19] Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.

[20] Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, 2014.

[21] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[22] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.

[23] John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

[24] Peter Gärdenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2004.

[25] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[26] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

[27] Dan Jurafsky and James H Martin. *Speech and language processing*. Pearson London:, 2017.

[28] András Kornai. The algebra of lexical semantics. In *the Mathematics of Language*, pages 174–199. Springer, 2010.

[29] András Kornai. Eliminating ditransitives. In *Formal Grammar*, pages 243–261. Springer, 2012.

[30] George Lakoff. *Women, fire, and dangerous things*. University of Chicago press, 2008.

[31] Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 270–280, 2015.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[33] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[34] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[35] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

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

[37] Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*, 2009.

[38] Davide Picca, Alfio Massimiliano Gliozzo, and Simone Campora. Bridging languages by supersense entity tagging. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 136–142. Association for Computational Linguistics, 2009.

[39] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM, 2011.

[40] Gábor Recski, Eszter Iklódi, Katalin Pajkossy, and Andras Kornai. Measuring semantic similarity of words using concept networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 193–200, 2016.

[41] Eleanor Rosch and Carolyn B Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, 7(4):573–605, 1975.

[42] Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large

scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[44] Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*, 2017.

[45] Morris Swadesh. Lexico-statistic dating of prehistoric ethnic contacts: with special reference to north american indians and eskimos. *Proceedings of the American philosophical society*, 96(4):452–463, 1952.

[46] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565, 2014.

[47] Jörg Tiedemann. Parallel data, tools and interfaces in opus. In *LREC*, volume 2012, pages 2214–2218, 2012.

[48] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[49] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, 2015.

[50] Hyejin Youn, Logan Sutton, Eric Smith, Cristopher Moore, Jon F Wilkins, Ian Maddieson, William Croft, and Tanmoy Bhattacharya. On the universal structure of human lexical semantics. *Proceedings of the National Academy of Sciences*, 113(7):1766–1771, 2016.

[51] Kai Zhao, Hany Hassan, and Michael Auli. Learning translation models from monolingual continuous representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1527–1536, 2015.