University of Ljubljana
Faculty of Computer and Information Science

Ljupche Milosheski

Cross–lingual mappings of contextual word embedding ELMo

BACHELOR'S THESIS

UNIVERSITY STUDY PROGRAMME
UNDERGRADUATE PROGRAMMES
COMPUTER AND INFORMATION SCIENCE

Mentor: prof. dr. Marko Robnik Šikonja

Ljubljana, 2019

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Ljupche Milosheski

Medjezikovne poravnave kontekstne vložitve besed ELMo

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: prof. dr. Marko Robnik Šikonja

Ljubljana, 2019

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Faculty of Computer and Information Science issues the following task:

Theme:

Word embeddings map individual words into a high-dimensional vector space, where similar words are close together. Since words from different languages are mapped similarly, they can be aligned in a common vector space, called cross-lingual embedding. Lately, contextual word embeddings map different meanings of a word into different vectors, depending on a word's context, typically the sentence in which the word appears.

For cross-lingual embedding of contextual word embedding ELMo, use the dictionary. Build a data set consisting of word translation pairs and contextually matching sentences. For the construction use words translations from English-Slovene dictionary and parallel corpora. Such a data set allows that standard approaches for supervised cross-lingual alignment are used for contextual embeddings. Test the method on the Slovene-English language pair. Analyse how the quality of cross-lingual embeddings depends on the size of the alignment training set.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Vektorske vložitve preslikajo posamezne besede v visokodimenzionalen vektorski prostor, kjer so podobne besede blizu skupaj. Ker se besede iz različnih jezikov preslikajo na podoben način, jih je mogoče poravnati v skupen prostor, kar imenujemo medjezikovne vektorske vložitve. V zadnjem času se uveljavljajo kontekstne vektorske vložitve, ki preslikajo različne pomene besed v različne vektorje, odvisno od konteksta, tipično stavka, v katerem beseda nastopa.

Za medjezikovno poravnavo kontekstnih vektorskih vložitev uporabite slovar. Zgradite podatkovno množico, sestavljeno iz parov prevodnih ustreznic in njihovih kontekstno ujemajočih se stavkov. Za gradnjo uporabite prevode besed in paralelni korpus. S tako množico je mogoče s klasičnimi pristopi medjezikovnih poravnav obravnavati tudi kontesktne vektorske vložitve. Metodo preskusite na paru slovenščina – angleščina. Analizirajte kakovost medjezikovnih vložitev v odvisnosti od velikosti učne množice za poravnavo.

Zahvaljujem se prof. dr. Marku Robinku Šikonji za mentorstvo pri diplomski nalogi. Njegovo znanje, nasveti in strokvno usmerjanje so bili dragoceni za nastajanju diplomskega dela ter za moje znanje.

За моjата фамилиjа
For my family

Contents

List of abbreviations

Abbreviation	Meaning
biLM	bidirectional language model
CBOW	continuous bag-of-words
CSLS	cross-domain similarity local scaling
LSA	latent semantic analysis
LSTM	long short-term memory neural network
ML	machine learning
MLM	masked language model
NLP	natural language processing
NN	neural network
SVD	singular value decomposition

Abstract

Title: Cross–lingual mappings of contextual word embedding ELMo

Author: Ljupche Milosheski

To work with textual data, machine learning algorithms, in particular, neural networks, require word embeddings – vector representations of words in high-dimensional space. There are languages with a small amount of available resources. Exploiting the knowledge from the well-resourced languages for under-resourced languages is possible with cross-lingual embeddings by aligning the embeddings of one language with the vector space of another language. Existing methods for aligning embeddings are intended for context-independent embeddings, where every word has one representation. We propose a method, based on a dictionary and a parallel corpus aligns contextual embeddings, which capture more information about the context in which words appear. The proposed method requires a small amount of bilingual data, which is available for many language pairs. We empirically show that the proposed method outperforms the baseline obtained by alignment of context-independent embeddings.

Keywords: cross-lingual word embeddings, contextual word embeddings, vector word embeddings, word translation, parallel corpus, vector space mappings, singular value decomposition.

Povzetek

Naslov: Medjezikovne poravnave kontekstne vložitve besed ELMo

Avtor: Ljupche Milosheski

Da bi algoritmi strojnega učenja, še posebej nevronske mreže, delali z besedilnimi podatki, potrebujejo vložitve besed — predstavitev besed v visokodimenzionalnem vektorskem prostoru. Za nekatere jezike je na voljo le majhna količina jezikovnih virov. Zanje je pomembno izkoriščanje znanja iz tehnološko bolj razvitih jezikov, kar omogočajo medjezikovne vložitve. Te vektorski prostor besed enega jezika preslikajo v vektorski prostor drugega jezika. Obstoječe metode za poravnavo vektorskih vložitev so namenjene kontekstno neodvisnim vložitvam, kjer ima vsaka beseda le eno predstavitev. Obstoječe metode za poravnavo vektorskih vložitev so namenjene kontekstno neodvisnim vložitvam, kjer ima vsaka beseda le eno predstavitev. Predstavljamo pristop, ki na podlagi slovarja in paralelnega korpusa poravna kontekstne vložitve, ki zajemajo več informacij o kontekstu, v katerem so besede uporabljene. Predlagani pristop zahteva majhno količino dvojezičnih virov, ki so na voljo za mnogo parov jezikov. Empirično pokažemo, da je predlagani pristop opazno boljši od izhodiščnega, s katerim poravnavamo kontekstno neodvisne vložitve.

Ključne besede: medjezikovne vložitve besed, kontekstne vložitve besed, vektorske vložitve besed, prevajanje besed, paralelni korpus, preslikave vektorskih prostorov, razcep z metodo singularnih vrednosti.

Razširjeni povzetek

Obdelava naravnega jezika je eno izmed ključnih področij računalnišva. Predstavitev besedilnih podatkov s števili omogoča algoritmom strojnega učenja delo z njimi. Tovrstni pristopi se imenujejo vložitve besed in preslikajo besede v visokodimenzionalne vektorske prostore realnih števil. Nekateri jeziki imajo na voljo omejeno količino jezikovnih virov, zato so modeli strojnega učenja za te jezike nezanesljivi. Vložitve besed imajo podobno strukturo v različnih jezikih. V diplomski nalogi izkoriščamo to podobnost za preslikavo med vektorskimi prostori besed. To potencialno omogoča, da prenesemo znanje modela iz enega jezika v drug jezik. Z našim pristopom, dobimo zanesljive vektorske vložitve z majhno količino dvojezičnih virov, ki je na voljo za mnogo parov jezikov. Ideja uporabljenega pristopa je, da z linearno transformacijo poravnamo vložitve enega jezika z vektorskem prostorom drugega jezika. Podobni pristopi so namenjeni poravnavi kontekstno neodvisnih vložitev. Mi pa smo jih prilagodili, da poravnajo kontekstne vložitve, ki zajemajo tudi informacij o kontekstu, v katerem so besede uporabljene. Kontekstne vložitve so zato uspešnejše v praktičnih problemih obdelave jezika. Kakovost razvitega pristopa smo ocenili na problemu prevajanja besed na paru slovenščina – angleščina.

Eden izmed prvih pristopov k vložitvam besed je latentna semantična analiza (LSA) [15], ki zgradi veliko in redko matriko. Njene vrednosti so sorazmerne številu pojavitev določenega izraza v danem dokumentu. Velikost matrike zmanjšamo z metodo razcepa po singularnih vrednostih (SVD),

hkrati pa dobimo vložitve besed.

Prvi uspešen pristop vložitev besed z nevronsko mrežo so predlagali Mikolov in sod. (2013) [18]. Predlagali so dve arhitekturi. Prva metoda Skip-Gram maksimizira verjenost sopojavitve besed, ki se v stavkih nahajajo blizu skupaj. Bolj formalno, naj bodo $b_1, b_2, \ldots, b_N$ besede, ki nastopajo v jeziku. Okolica besede b_i je $C_i = \{b_j \mid 0 <\mid i - j \mid \leq k\}$, kjer je k velikost okolice. Model poskuša maksimizirati kriterijsko funkcijo

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{x \in C_i} \log p(x|b_i),$$

kjer je $p(x|y)$ verjetnost pojavitvi besede x pri dani besedi y [7]. Druga arhitektura se imenuje Continuous-bag-of-words in je nasprotna prvi, kar pomeni, da je za dano besedo cilj maksimizirati verjenost pojavitve besede med bližnjimi besedami t.j. maksimiziramo kriterijsko funkcijo

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{x \in C_i} \log p(b_i|x).$$

Težava teh dveh arhitektur je, da ima vsaka beseda le eno predstavitev, ne glede na to, da imajo besede več pomenov.

Mikolov in sod. (2013) [19] so opazili, da imajo vložitve besed različnih jezikov podobno strukturo. To lastnost so izkoristili za poravnavo vektorskih prostorov različnih parov jezikov. Njihova kriterijska funkcija minimizira evklidsko razdaljo med pari prevodnih ustreznic. Kasneje je nekaj drugih raziskovalnih skupin poravnavo še izboljšalo. Xing in sod. (2015) [33] so maksimizirali kosinusno podobnost med prevodnimi pari ter dodali omejitev, da mora biti matrika, ki preslika vložitve ortogonalna. Artexte in sod. (2016) [3] so problem prevedli na reševanje ortogonalnega Prokrustovega problema. Problem pravi, da za dani matriki S in T iste velikosti iščemo ortogonalno matriko Q, tako da je QS najboljši približek matrike T. Bolj formalno, matrika Q je definirana kot

$$Q = \underset{X}{\operatorname{argmin}} \|XS - T\|_F,$$

kjer je $Q \in \mathbb{R}^{N \times N}$ in $Q^\intercal Q = QQ^\intercal = I$. Vsi pristopi za poravnavo vložitev besed so namenjeni kontekstno neodvisnim vložitvam.

Peters in sod. (2018) [21] so predlagali arhitekturo kontekstnih vložitev besed ELMo (Embeddings From Language Models), kjer je predstavitev vsake besede odvisna od konktesta, v katerem je ta uporabljena. Arhitektura ima tri glavne komponente: nevronsko mrežo z dolgim kratkoročnim spominom (angl. long short-term memory neural network, LSTM) za napovedovanje besed, ki sledijo dani besedi, LSTM mrežo za napovedovanje predhodnjih besed in konvolucijsko nevronsko mrežo na znakih za kontekstno neodvisno predstavitev besed. Kontekstne vložitve besed vsebujejo več informacij kot kontekstno neodvisne vložitve. Empirično so pokazali, da je ELMo uspešna vložitev na šestih problemih obdelave naravnega jezika.

V delu razvijamo pristop, ki poravna kontekstualne vložitve besed. Potrebujemo pomnilnik prevodov (angl. translation memory), ki je sestavljen iz stavkov v enem jeziku in ustrezenih prevodov v drugem jeziku. Ideja predlaganega pristopa je z uporabo slovarja zgraditi paralelni korpus, ki je sestavljen iz parov prevodnih ustreznic in njihovih kontekstno ujemajočih se stavkov. S tako množico je mogoče s klasičnimi pristopi medjezikovnih poravnav obravnavati tudi kontekstne vektorske vložitve. V našem primeru smo uporabili pomnilnik prevodov dokumentov Evropske Unije [30], pristop pa smo ocenili na paru slovenščina – angleščina. Kontekstne vložitve smo dobili z že naučenima modela ELMo. Poravnave pa smo izračunali z knjižnico MUSE [8].

Predlagan pristop smo ovrednotili na problemu prevajanja besed, kjer za podane besede v enem jeziku poiščemo čimveč pravilnih prevodov v drugem jeziku. Za podano konstanto n je poizvedba pravilna, če se vložitev pravilnega prevoda besede nahaja med n sosedi vložitve preslikane poizvedbe. Za mero razdalj med vložitvami smo uporabili kosinusno podobnost ter metriko CSLS, ki so jo predlagali Conneau in sod. (2017) [8].

Analizirali smo dva glavna scenarija. Ker ima lahko ena beseda v is-

tem konktestu več predstavitev z modelom ELMo, smo najprej poiskali par predstavitev, ki da po poravnavi najboljše rezultate. Kot drugo nalogo smo analizirali kakovost poravnave v odvisnosti od velikosti učne množice. Pri prvem scenariju smo najboljši rezultati dobili, ko poravnamo predstavitve druge plasti obeh modelov ELMo. To je logično, saj je druga plast zadnja in vsebuje največ informacij o besedi. Pri drugi analizi smo s povečevanjem učne množice ugotovili, da se poveča tudi kakovst poravnave. Naš pristop je bistveno boljši od izhodiščnega, pri katerem poravnavamo kontekstno neodvisne vložitve. Rezultati analize so v tabeli 1. Ker so kontekstualne vložitve boljše od kontekstno neodvisih vložitev, dobimo s predlaganim pristopom možnost za kakovosten prenos modelov strojnega učenja v jezike z omejenimi viri, saj pri učni množici velikosti 20.000 besed dobimo precej dobre rezultate. Na sliki 1 prikažemo nekaj neporavnanih vložitev, ki so del testne množice, na sliki 2 pa prikažemo poravnane vložitve iz slike 1. Poravnava ni popolna, ker je lahko dejanska preslikava nelineara.

Metoda \ Vel.		500	1,000	5,000	10,000	20,000	35,000	50,000
tok slyr-slyr	1NN	8.3	16.5	31.5	36.0	37.1	40.3	**41.1**
	5NN	19.7	33.1	53.1	55.5	57.9	60.3	**60.8**
	10NN	28.3	41.9	60.3	62.4	65.6	**66.9**	66.7
	1CSLS	21.9	33.3	47.2	49.3	49.9	**51.2**	50.7
	5CSLS	39.5	54.1	66.4	66.4	**69.3**	68.8	69.1
	10CSLS	50.7	61.3	72.0	73.6	74.7	**74.9**	74.7
tok fstx-fstx	1NN	19.9	23.9	33.5	31.9	35.6	**36.3**	35.5
	5NN	25.1	31.1	41.0	42.2	45.4	46.2	**47.0**
	10NN	28.7	32.7	45.0	47.0	49.4	49.4	**51.0**
	1CSLS	20.3	25.1	32.7	32.3	34.7	**35.1**	34.7
	5CSLS	24.7	32.3	39.0	39.8	40.6	42.6	**43.4**
	10CSLS	26.7	34.7	42.2	44.6	43.4	47.4	**48.2**

Table 1: Natančnost (v %) na problemu prevajanju besed pri različnih velikostih učne množice. Okrajšave "fstx" and "slyr" predstavljajo kontekstno neodvisne vložitve oz. kontekstne vložitve druge plasti modela ELMo.

Figure 1: Primer neporavnanih vložitev besed v slovenščini in angleščini. Sliko smo dobili z metodo glavnih komponent (angl. principal component analysis, PCA).

Figure 2: Poravnane vložitve besed prikazane na Sliki 1. Sliko smo dobili z metodo glavnih komponent.

Chapter 1

Introduction

Computer understanding of natural language is an important task for humans. The amount of textual data on the Internet is enormously increasing every year. There are various tasks that benefit from natural language understanding such as finding relevant queries and documents, machine translation, evaluation of documents, named entity recognition, etc. Finding patterns in unstructured data is crucial for humans to accomplish their objectives. Yet it is a tedious and infeasible task when the amount of data is gigantic. Nowadays, the best approach to solving natural language tasks is with word embeddings. Word embeddings are machine learning (ML) techniques that represent words in a high-dimensional vector space. ML algorithms are known to benefit from huge amount of data existing for some languages. However, there are languages with limited resources. Exploitation of knowledge from the well-endowed languages in under-resourced languages is possible with cross-lingual embeddings.

Cross-lingual embeddings map word embeddings of one language into the vector space of another language. This is possible since the structure of word embeddings across distinct languages is preserved. There are a few methods that align the embeddings of two or more languages in a supervised and unsupervised manner. The supervised alignment problem has been solved to some

extent. The problem can be translated to the orthogonal Procrustes problem, which has a unique solution in closed-form. The orthogonal Procrustes problem states that for given two matrices A and B of same dimensions, find an orthogonal matrix Q such that QA best approximates B. The unsupervised methods try to model the problem as an adversarial game or exploiting the distributional information of words.

All cross-lingual alignment methods are originally meant for alignment of context-independent embeddings. With small modifications of the supervised methods, we present a method for cross-lingual embedding of contextual embeddings. Cross-lingual embeddings intuitively capture more information about words, since words in different contexts may have different meanings.

Results presented in this analysis are evaluated on word translation retrieval task. The problem states that for a given set of query words from a source language, the objective is to find as many correct translations from a set of words in a target language. The results are compared to results of other researchers and to a baseline obtained with the alignment of context-independent embeddings.

The work is composed of six chapters. In Chapter 2, a brief introduction and explanation of what was done in the areas of word embeddings and their alignment are presented. In Chapter 3, the process of obtaining embeddings and training and evaluation datasets is described. A formal definition of the cross-lingual alignment problem and explanation of the evaluation methods are present in Chapter 4. Chapter 5 consists of evaluation scenarios and results. In Chapter 6, we summarize and present the strengths and weaknesses of the proposed method as well as ideas for future work.

Chapter 2

Background

In this chapter, we present previous work in the fields of word embeddings and alignment of embeddings. The first section covers what was done in the field of context-independent and contextual word embeddings. The second section contains an overview of supervised, semi-supervised, and unsupervised methods for cross-lingual embeddings.

2.1 Word embedding techniques

Word embeddings represent multiple techniques that map each word to a high-dimensional vector of real numbers. One of the first such technique is called latent semantic analysis (LSA) [15]. It uses a term-document matrix which entries to express the occurrences of terms in documents. An example of weighting is a term frequency-inverse document frequency. The weight of each element is proportional to the number of times the term appeared in each document where rare terms have a higher weight to reflect their importance. Because the term-document matrix is large and sparse, LSA reduces its dimensions with singular value decomposition (SVD). Vector representation of documents and terms in the reduced latent space represent embeddings. LSA the most used method before the introduction of word

embeddings with a neural network (NN) by Mikolov et al. [18, 20] in 2013. The neural embeddings replaced LSA because they require less resources.

A naive way for word representation in natural language processing (NLP) is a one-hot representation of words where each word is represented by a high-dimensional vector that has all of its components equal to 0 except for one component that is set to 1. There are a few reasons why word embeddings replaced the one-hot representation.

1. Vectors in one-hot representation are extremely high-dimensional, sparse and their size increases with the vocabulary size. The higher the dimension is, the more resources (time and space) we use for each word's representation. On the other hand, word embeddings preserve the number of dimensions which may be fixed for a certain problem.

2. ML algorithms that use one-hot representations are prone to overfitting because of their high-dimensional representation. Curse of dimensionality [6] is a notorious problem that many algorithms are prone to. The algorithms are likely to focus on random noise that looks like a pattern, but actually is not. The predictions of algorithms that learn noisy patterns will not be relevant. Word embeddings are more resistant to this issue because they use a lower-dimensional representation of fixed length.

3. One-hot representation does not contain any information about how two words are related. Ideally, we would like the vector representation of any pair of words to reveal how similar or distinct the given words are. With word embeddings, we can compute a cosine similarity between two words to find out how close the two words are.

Mikolov et al. [18] were the first to introduce the word embeddings trained with NN. An important characteristic of this method is that it is unsupervised. That means we can train it on raw textual data. The similar-

ity between pairs of semantically close words is high and also distances between semantic meanings is preserved. For instance, $v("king") - v("man") + v("woman")$ is a vector that is closest to $v("queen")$, where $v(x)$ is vector representation of the word x. Even though the training of NN is time-consuming, the process can be sped up by distributing the process of learning [20]. It is also interesting how two seemingly distinct problems turned out to be much closer than imagined. Levy et al. [16] showed that the training of NNs during word embeddings perform weighted matrix factorization of the shifted pointwise mutual information matrix, which is closely related to the matrix factorization technique used for recommendation systems [13] and LSA [15].

Mikolov et al. [18] proposed two architectures for learning word embeddings. The first one is called skip-gram model where we are trying to predict the words preceding and succeeding a given word. Therefore, for a given sequence of words $w_1, w_2, ..., w_N$ our objective is to maximize the average log-likelihood

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{x \in C_i} \log p(x|w_i),$$

where k is the size of the training context and $C_i = \{w_j \mid 0 <| \, i - j \, | \leq k\}$, where $j \in \mathbb{N}$ and k is the size of the training context [7]. Figure 2.1 shows the architecture of the model.

The second proposed architecture is called continuous bag-of-words (CBOW) and is complementary to the previous model. In this model, we are learning to predict a word from a given sequence of preceding and succeeding words. For the given sequence of words $w_1, w_2, ..., w_N$, our criterion function is to maximize the average log-likelihood

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{x \in C_i} \log p(w_i|x),$$

where k is the size of the training context and $C_i = \{w_j \mid 0 <| \, i - j \, | \leq k\}$, where $j \in \mathbb{N}$ and k is the size of the training context. Figure 2.2 shows the architecture of the model.

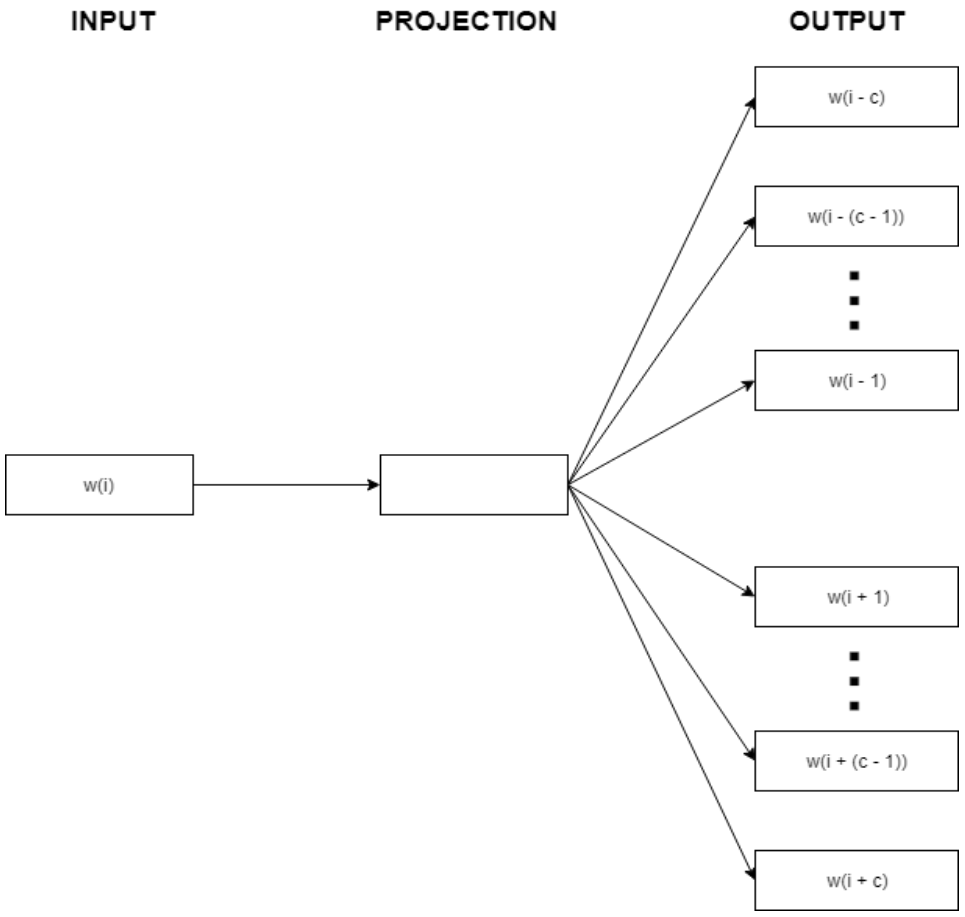

Figure 2.1: Skip-gram architecture predicts surrounding words given the current word.

There are languages such as Turkish and Finnish, where words are made of multiple morphemes. The words in these languages have various forms and therefore, the morphemes contain important information about the context. Previous word embedding approaches can be significantly improved for such languages if they take into consideration the morphemes. To solve this issue, Bojanowski et al. [7] introduced an extension to the CBOW model proposed by Mikolov et al. [18, 20], where instead of consecutive words, the model takes into consideration consecutive n characters. They called the model bag of

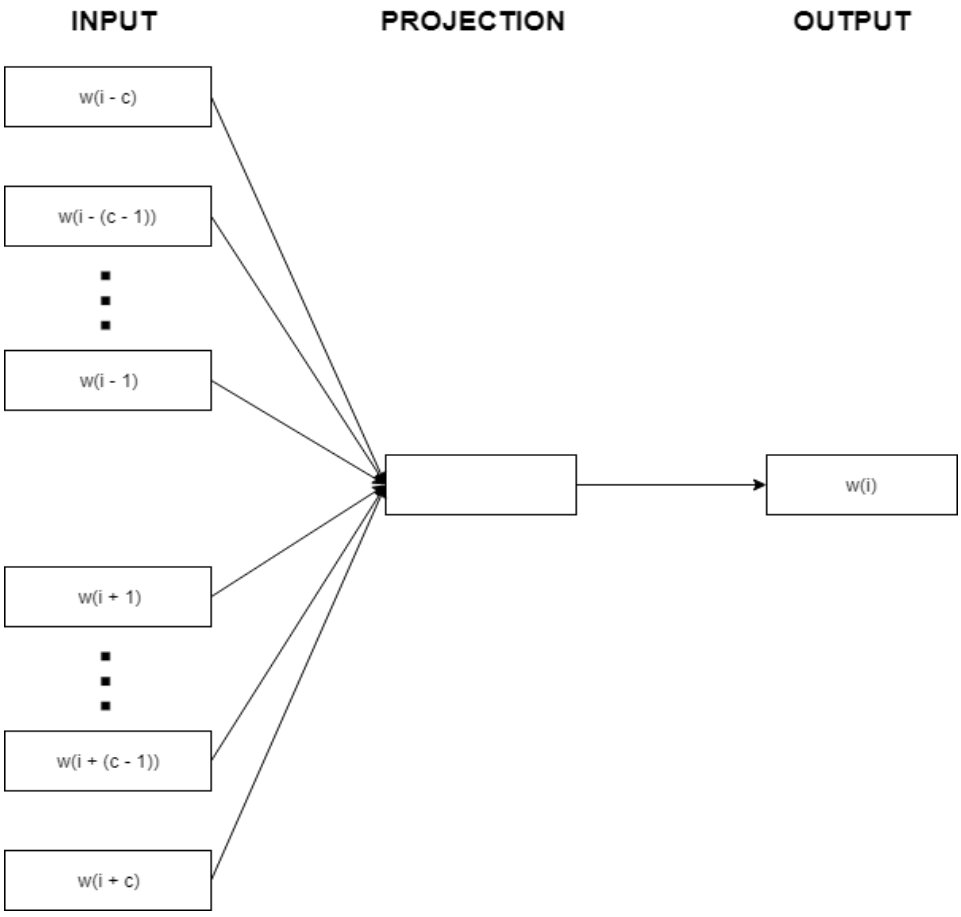

Figure 2.2: The CBOW architecture predicts the current word based on the context.

character n-grams. It was empirically shown, that this model outperforms its predecessor.

Researchers tried to make better use of context in which a given word appears. In other words, they wanted to find an embedding technique such that the embedding of a word would be dependent on the entire context in which it was used. This type of embeddings is called contextual embeddings. It relies on the hypothesis by Harris [12], which states that words that occur in similar contexts are likely to have similar meanings. Peters et al. [21]

created an architecture that uses the entire context in which the word appears. It became state-of-the-art for six NLP tasks. They called it deep contextualized word representations. It is deep because all internal layers of biLM (bidirectional language model) affect the result. Since each representation is computed from multiple layers, each consisting of a model, they named it Embeddings from Language Models (ELMo). In particular, L-th layer consists of $2L+1$ word representations, L of which are impacted by the preceding words, L by the succeeding words and 1 is context-independent embedding that is calculated via token embeddings or a convolutional neural network over characters.

Bidirectional Encoder Representations from Transformers (BERT) is the current state-of-the-art architecture for word embeddings. It was introduced by Devlin et al. [9] and is based on the encoder from the transformer architecture proposed by Vaswani et al. [32]. Unlike other models that are based on the transformer architecture like OpenAI GPT that uses left-to-right architecture [22], BERT also uses the tokens that appear on the right side. Moreover, BERT is a masked language model (MLM) meaning that some of the input tokens are hidden and its objective during the training phase is to predict them. It is inspired by Cloze test [29] because MLM's objective is measured by how good it performs on the Cloze test. An intuitive explanation of why this model works is that it combines both, the left and the right context, and the contextual information is crucial to obtain good results. It was empirically proven that BERT is very successful in eleven NLP tasks [9].

2.2 Alignment of cross-lingual embeddings

Embeddings trained on one language can be mapped into the vector space of another language. Supervised, semi-supervised and unsupervised techniques that align cross-lingual embeddings were proposed. Many of them use a linear transformation to map words from source space into a target

space. Supervised methods' criteria function can be translated to the Procrustes problem. There are more ideas for unsupervised alignments, such as adversarial training or using the distributional information of the words.

Mikolov et al. [19] were the first to notice that even the word embeddings in distinct languages like English and Vietnamese have a similar structure. They exploited this property and made a dictionary of 5,000 words that they used as anchor points to learn a linear transformation. The result was evaluated on a word translation retrieval task. Their criteria function minimized the sum of squared Euclidean distances for pairs in the dictionary. Xing et al. [33] improved it by normalizing the embeddings, adding the orthogonality constraint and maximizing the cosine similarity between dictionary entries. The proposed orthogonality constraint enforces that the distances between mapped embeddings are preserved. Artetxe et al. [3] successfully solved the problem by translating it to the orthogonal Procrustes problem, and adding a few logical and intuitive modifications to the previous techniques. At this point, the results obtained with the supervised methods were deemed sufficiently good.

Conneau et al. [8], Lample et al. [14] and Zhang et al. [34] solved the problem in an unsupervised manner by modeling it as an adversarial game. In that game, one player is trying to distinguish whether a given word is from the source or target space. The solution of the problem is found by the second player who is training a linear transformation that maps the source vectors into the target space. The quality of the solution depends on the result of the second player. In addition to the adversarial game, a refinement procedure was proposed that can be used after the initial transformation has been found. During the adversarial game, we can create a bilingual dictionary by checking how small modifications in the linear transformation affect the result. The entries in this induced dictionary can then be used as anchor points that should be aligned. The alignment is similar to the supervised methods for the Procrustes problem. Overall, this modification significantly

improved the results and they were almost as precise as of the supervised methods.

Artetxe et al. [4] proposed a semi-supervised iterative semi-supervised self-learning model that starts with a small dictionary whose size increases with the training. This approach could easily get stuck in poor local optima, especially when the initial solution was not good enough. The current most precise results with unsupervised methods were achieved by Artetxe et al. [5] who proposed an approach that is based on the distributional information of the words. It is based on the observation that two equivalent translations have more similar distributions than two unrelated words.

Chapter 3

Dataset

This chapter covers the phases of processing the data from its raw form until the words are embedded. We used the raw translation memory data from the official European Union's documents. Only sentences for Slovene and English were processed. The aligned sentences were first tokenized and then lemmatized. With a bilingual dictionary, a parallel corpus was made from the lemmatized sentences. Finally, each word's lemma and its original form were contextually embedded.

3.1 Processing raw data

A raw TM data was obtained from the OPUS web page [30] in Translation Memory eXchange format - a special XML format. It contains many European Union's legislative documents in 24 EU languages. This dataset was made by Tiedemann [31]; however, it is based on the dataset published by Steinberger et al. [28] when the European Commission's Directorate-General for Translation has made its multilingual Translation Memory for the Acquis Communautaire, DGT-TM, publicly accessible. Further updates and statistics were later released by Steinberger et al. [27, 26].

Translation memory is a document that consists of many sentences, each

aligned with its translations into multiple languages. Only sentences in Slovene and English were chosen, as only data for these two languages were required for our analysis. The resulting dataset contained almost 2M distinct parallel sentences in Slovene and English.

3.2 Creating a parallel corpus

The result from the previous phase was a TM for Slovene and English. The next goal was to make a parallel corpus – a file that consists of multiple lines, each of which contains a pair of words from both languages that appear in the same sentence in the TM with the restriction that one word is a direct translation of the other.

In order to build the parallel corpus, we need to find if a translation of a word in one language appears in the other. For that reason, it is required that the words in the sentences of the TM are lemmatized. Since lemmatization is time-consuming, only 3.5% or about 70,000 sentences from the TM were randomly chosen. The sentences in Slovene were tokenized and lemmatized with tokenizer and tagger from the ReLDI project [17] with source code on GitHub [1]. WordNet [10] tokenizer and lemmatizer were used for the sentences in English.

Having the sentences lemmatized, we constructed a parallel corpus. A dictionary is required in order to look up for one word's translation in the other language. Center for Language Resources and Technologies of University of Ljubljana provided pairs of words from the Oxford Slovene-English dictionary. The resulting parallel corpus was made by checking if a pair of lemmas from a pair of sentences appears in the dictionary. If so, the entry was added to the parallel corpus. The parallel corpus contained about 700,000 pairs of tokens[1]. That means that for each sentence in one language, we found on average 10 words with an appropriate translation in the other

[1]Lemmas of words or words in original forms – as they appear in the sentences.

language. Phrasal expressions were excluded from the process.

3.3 Embedding the tokens from the parallel corpus

Since the objective of our analysis is to find, evaluate and analyze how good contextual cross-lingual embeddings are, pre-trained ELMo models [21] were used for each language. The models were not fine-tuned for our task. Tokens from the parallel corpus were embedded in the context of the sentence in which they appear. Both, the original form and its lemma were embedded in order to evaluate how important the original form is. In addition, pre-trained context-independent embeddings with the FastText method [7] were obtained from the dataset published by Grave et al. [11]. Their alignment was a baseline for evaluating the quality of the alignment of contextual embeddings.

The ELMo model for Slovene was obtained from the Laboratory for Cognitive Modeling, University of Ljubljana. Its output had 3 layers: context-independent representation, contextual forward LSTM (long short-term memory neural network) and contextual backward LSTM. A vector that is point-wise average of all components was computed for the previous layers. In the end, there were 4 representations for every token, each with dimension 1024.

The ELMo model for English has been trained by Google and is publicly available as part of TensorFlow.[2] It had a layer that outputs context-independent embeddings with dimension 512, contextual forward LSTM with dimension 1024, contextual backward LSTM with dimension 1024 and a weighted sum of the previous layers with dimension 1024. The context-independent embeddings were dropped to avoid dimension incompatibility with Slovene model. A vector that is point-wise average of all components of the forward and backward LSTM representations was computed. There

[2]https://tfhub.dev/google/elmo/2

were again 4 representations for every token, each with dimension 1024.

Because there was a dimension mismatch in the context-independent embeddings of the English and Slovene ELMo model, new context-independent embeddings were obtained from a pre-trained dataset with the FastText method [11]. Their dimension was 300. Results of this alignment served a baseline in comparison of alignment for contextual embeddings.

Since the contextual embedding procedure is time-consuming, only 65,000 randomly chosen entries from the parallel corpus were chosen. Moreover, 269 of them were dropped because context-independent embeddings were not available for either the original form or lemmas of some tokens. Therefore, the final dataset contained 64,731 tokens in each language, each of which had 4 ELMo and 2 FastText embeddings.

Chapter 4

Methodology

This chapter contains a formal definition of the cross-lingual alignment problem and how it can be solved, as well as an explanation of the methods used to evaluate the solutions.

4.1 Formal definition of the problem

Let denote $X^{(i)}$ the i-th column of matrix X and $X_j^{(i)}$ be the j-th element in $X^{(i)}$. Let us define the operator $\|\cdot\|_F$ which represents Frobenius norm computed as

$$\|X\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|X_j^{(i)}|^2},$$

for any $X \in \mathbb{R}^{n\times m}$. Let us assume that we have already computed the word embeddings as described in Chapter 3, that there are M instances[1] in each language and the embedding dimension is N. Thus, there are M embeddings for each language represented by a vector in $\mathbb{R}^N$. Let S and T be two matrices in $\mathbb{R}^{N\times M}$ such the $S^{(i)}$ and $T^{(i)}$ represent an embedding of a certain token from the first language and its corresponding translation in the

[1] In other words tokens.

second language, respectively, for $i \in \{1, 2, \ldots, M\}$. We call such matrices embedding matrices.

Our goal is to find an orthogonal matrix Q so that QS best approximates T. Formally, the matrix Q is defined as

$$Q = \underset{X}{\operatorname{argmin}} \|XS - T\|_F$$

subject to $Q^\intercal Q = QQ^\intercal = I$ such that $Q \in \mathbb{R}^{N \times N}$. This problem is also known as the orthogonal Procrustes problem. Schönemann [24] was the first to find a solution using SVD. The solution is $Q = UV^\intercal$ where $TS^\intercal = U\Sigma V^\intercal$ is the SVD decomposition of $TS^\intercal$. The solution is in closed-form and is unique since the SVD of a matrix is unique. If the orthogonality constraint on Q did not exist, the problem could be solved by computing Moore-Penrose pseudoinverse. However, enforcing the orthogonality constraint significantly improves the results as shown by Xing et al. [33].

4.2 Evaluation methods

In this section, we present two evaluation methods for the quality of cross-lingual embeddings. Their understanding is crucial for understanding the results and making the conclusions. We continue to use the definitions from Section 4.1.

Results were evaluated on the word translation retrieval task. The problem states that for a given set of query words from a source language, the objective is to find their correct translations of the in a target language. The task can be solved with cross-lingual word embeddings. We first map a source word embedding to the target space. The mapping of a query word is successful if the mapped source word is within the first n neighbours of the correct translation in the target space.

4.2.1 Nearest neighbours

Let us define a n neighbourhood $N(n, s, t, S, T, Q)$ for constants n denoting the size of neighbourhood, s denoting the number of embeddings in the source space, t denoting the number of embeddings in the target space, an embedding matrix S in $\mathbb{R}^{N \times s}$, an embedding matrix T in $\mathbb{R}^{N \times t}$, and a linear mapping that we evaluate Q in $\mathbb{R}^{N \times N}$ such that:

1. $n \leq t$, and
2. $\mid N(n, s, t, S, T, Q) \mid = n$, where $\mid \cdot \mid$ denotes size of a set, and
3. $N(n, s, t, S, T, Q) \in \mathcal{P}(\{1, 2, \ldots, t\})$, where $\mathcal{P}(\cdot)$ denotes powerset of a set, and
4. $\forall i \in \{1, 2, \ldots, t\}\, \forall j \in N(n, s, t, S, T, Q) : d(T'^{(s)}, T^{(i)}) \leq d(T'^{(s)}, T^{(j)}) \Rightarrow i \in N(n, s, t, S, T, Q)$, where $T' = QS$ and $d(u, v)$ is a represents the distance between the vectors u and v.

We also define a function that evaluates whether a mapping of a query token is successful. For a constant i denoting the column of the query in the source embedding matrix, $I(n, i, s, t, S_{evl}, T_{evl}, Q)$ is computed as

$$I(n, i, s, t, S_{evl}, T_{evl}, Q) = \begin{cases} 1, & \text{if } i \in N(n, s, t, S_{evl}, T_{evl}, Q); \\ 0, & \text{otherwise.} \end{cases}$$

Function I denotes if the mapping of a source word is successful, i.e., if the correct translation of the word is within the first n neighbours of the mapped embedding.

Let us suppose that we have already obtained the matrix Q described in section 4.1. Its quality can be measured on an evaluation dataset represented by S_{evl} in $\mathbb{R}^{N \times s}$ and T_{evl} in $\mathbb{R}^{N \times t}$, a source and target embedding matrices respectively. The precision on the word translation retrieval task is then

computed as

$$NN(n, s, t, S_{evl}, T_{evl}, Q) = \frac{1}{s} \sum_{i=1}^{s} I(n, i, s, t, S_{evl}, T_{evl}, Q).$$

4.2.2 Cross-domain similarity local scaling

In every language, there are words that are more frequent than others. Embedding them into a high-dimensional space might cause a problem for the nearest neighbours method because common words tend to form hubs, i.e., they have many neighbours. For these words, the probability of being the nearest neighbour of other points is high. Cross-domain similarity local scaling (CSLS) is a method that mitigates this issue.

The nearest neighbours relationship is by definition asymmetric. If x is among the first n nearest neighbours of y, it does not imply that y is among the first n nearest neighbours of x. In high-dimensional spaces, this leads to a phenomenon that some vectors called hubs, are with high probability nearest neighbours of many other points, while separated points are not nearest neighbours of any point as shown by Radanović et al. [23].

To mitigate the hubness problem, Conneau et al. [8] proposed a new method called CSLS that decreases the similarity of hubs. For a mapped source embedding s and target embedding t, CSLS first computes the average cosine similarity of s and t for their n nearest neighbours denoted by $r_T(s)$ and $r_S(t)$ in the other language, respectively. $CSLS$ is computed as

$$CSLS(s, t) = 2cos(s, t) - r_T(s) - r_S(t),$$

where $cos(\cdot, \cdot)$ is cosine similarity. Intuitively, the method decreases similarity associated with vectors lying in dense areas. Conversely, it increases it for isolated vectors. For a given source embedding, CSLS can be used as a method to calculate the similarity between it and the embeddings in the target space. A query in the word translation retrieval task is successful

with the nCSLS method if the correct translation from the target space is one of the n most similar embeddings to the mapped source embedding.

Chapter 5

Evaluation scenarios

In this chapter, we empirically show that the proposed method outperforms the baseline – alignment of context-independent embeddings. In addition, we obtained quite good results compared to methods proposed by other researchers evaluated on different languages. We first describe the testing scenarios, then we present the results and compare them with the results of other researchers. Last, we give reasonable explanations for the obtained results.

5.1 Scenarios

We present two main analyses. We first analyze different mappings between embedded spaces in order to choose the best mapping. Then we analyze how the size of the training dataset affects the results. The results are evaluated on the word translation retrieval task with the methods described in Chapter 4. Embeddings were aligned in a supervised way, by finding the solution to the orthogonal Procrustes problem. The MUSE library[1] proposed by Conneau et al. [8] was used to perform all calculations. The refinement procedure available in the library was disabled in our analyses since it is intended for

[1] https://github.com/facebookresearch/MUSE

unsupervised method. We tried to use it with the supervised method, but it always degraded the performance for training datasets larger than 1,000.

From the final dataset of 64,731 tokens described in Chapter 3, 50,375 randomly chosen tokens were used to represent the source and target embedding space. The chosen tokens did not change across scenarios. An evaluation dataset of 375 tokens was created from them. Some of the remaining 50,000 tokens were randomly chosen for the training datasets. The size of the evaluation dataset was chosen in such a way that the proportion between the number of queries and embeddings in the target space is same as in the datasets proposed by Conneau et al. [8], which contain 1,500 queries and 200,000 embeddings. The things that change across testing scenarios are: type of tokens (original form or lemmas of words), the embedding method (4 ELMo and 2 FastText methods) and the size of the training datasets.

The goal of the first analysis is to choose the best mapping. As described in Section 3.3, there are 2 types of tokens and 6 different embeddings for each language. Because of dimensions mismatch between contextual and context-independent embeddings, there are at most 68 possible mappings, i.e., there are 34 unidirectional mappings, consisting of 32 contextual mappings (we have 2 types of tokens, 4 contextual embeddings in the source language and 4 contextual embeddings in the target language), and 2 context-independent mappings (we have 2 types of tokens). There are 68 possible mappings for both directions, Slovene-English and English-Slovene. We evaluated only unidirectional mappings where Slovene is the source language and English is the target language. However, some of the mappings do not make sense. For instance, mapping context-independent Slovene embeddings in the contextual English embeddings space is expected to give bad results. For simplicity, the results of such mappings are excluded, and we present the 26 most successful mappings. For this analysis, the embeddings were trained on a fixed dataset of 5,000 tokens.

In the first analysis, we used a fixed dataset to avoid random fluctuations

in results since the goal was to find the best mapping. The second analysis checks how the size of the training dataset affects the results. We evaluated it only for the best mapping from the first analysis. First, training datasets of various sizes were randomly created, then they were evaluated on the evaluation dataset of 375 tokens.

5.2 Results

We present the results from the first and second analysis. The alignment procedure and results were computed with the supervised method from the MUSE library, proposed by Conneau et al. [8].

As we can see in Table 5.2 and Table 5.3, the second layer embedding of the Slovene model gives the best results for both types of tokens. When aligning the original forms, the best precision is obtained when it is paired with the second layer or average embedding of the English model. When aligning lemmas, only the second layer embedding of the English model should be used. In both scenarios, context-independent embeddings perform well for small neighbourhoods. That suggests that the alignment of contextual embeddings better approximates hubs rather than single embeddings. It is also logical that the contextual embeddings align better with the original forms of words whereas the alignment of context-independent embeddings favors lemmas.

When the dataset size increases, the precision also increases, as we can see in Table 5.4, Figure 5.3 and Figure 5.4. It is again true that for small neighbourhoods alignment of context-independent embeddings of lemmas performs better. If we consider larger neighbourhoods, alignment of contextual embeddings of words in their original forms outperforms it. It should be noted the two results are not quite the same because when aligning context-independent embeddings of lemmas, we only retrieve lemmas. These lemmas cannot be used for word translation, as they are rarely the correct translation since

many word forms may have the same lemma. In some cases, after the dataset size reaches 20,000, the precision occasionally decreases. The reason for that might be that the real transformation between languages is not linear and we are approximating it with a linear transformation. The same appeared in Aldarmaki and Diab's work [2], who got fluctuations in the precision as the dataset increased.

Figure 5.1: Two dimensional PCA showing unaligned Slovene and English embeddings. Both embeddings are outputs of the second layer of the ELMo models described in Section 3.3.

To illustrate our results, we produced Figure 5.1 and Figure 5.2. The first figure shows embeddings in Slovene and English obtained by the output of the second layer in the ELMo models described in Section 3.3. We clearly see that embeddings in both languages are quite apart. The second figure shows a mapped version of the Slovene embeddings into the vector space of the English embeddings. A linear mapping was learnt on a training dataset

Figure 5.2: Two dimensional PCA showing aligned version of the embeddings in Figure 5.1. A linear transformation was trained on a training dataset with 50,000 entries. The tokens shown in the picture were not included in the training dataset.

with 50,000 entries. The presented tokens were not included in the training dataset. Every Slovene word is closely mapped to its correct translation. The illustrated mapping is not perfect since the real transformation may not be linear. In addition, PCA shows only 2 out of 1024 dimensions in the original space.

All things considered, our results are reasonably good. We empirically showed that our method gives more precise word retrievals than context-independent embeddings. If we compare the precision of our results to the ones obtained Aldarmaki and Diab [2] presented in Table 5.1, we got slightly worse results. There are several reasons for that. First, they trained their own ELMo model on the same dataset for multiple languages, while we used two ELMo models that were trained on totally different datasets. As shown

sl-en		de-en		es-en	
1NN	5NN	1NN	5NN	1NN	5NN
41.1	60.8	50.0	63.5	56.5	70.9

Table 5.1: Comparision between the precision of our method and precision obtained by Aldarmaki and Diab [2] for German and Spanish embeddings.

by Conneau et al. [8], such approach leads to significantly worse results. Second, it is expected that the alignment between two similar languages gives better results. German and Spanish are languages that are closer to English than Slovene. Schuster et al. [25] evaluated cross-lingual alignments for multiple languages. When using the supervised method of the MUSE library, they got a difference of 18% in precision between the alignment of French and Swedish in English word retrieval task. This suggests that the alignment of closer languages gives higher precision and that our results are pretty good. Last, Aldarmaki and Diab used much larger training dataset of 1M sentences, which results in several million tokens. We used a training dataset of at most 50,000 tokens. Since the precision is increasing, it would not be surprising if our results give similar or better precision if trained on that similarly large dataset.

Method / sl-en emb. type	1NN	5NN	10NN	1CSLS	5CSLS	10CSLS
fstx-fstx	33.5	41.0	45.0	32.7	39.0	42.2
flyr-flyr	24.0	42.1	50.1	33.3	54.7	61.1
flyr-slyr	16.3	40.3	48.8	32.8	55.5	62.7
flyr-elmo	21.6	42.7	51.5	34.4	56.3	64.0
flyr-avg	23.2	44.3	52.0	36.5	56.8	63.3
slyr-flyr	26.4	43.7	49.6	36.3	56.0	63.5
slyr-slyr	31.5	**53.1**	60.3	47.2	**66.4**	**72.0**
slyr-elmo	32.7	52.0	60.3	46.9	64.5	70.7
slyr-avg	**34.4**	52.3	60.0	47.2	64.5	70.4
avg-flyr	25.3	44.0	51.2	36.8	56.3	63.7
avg-slyr	27.7	51.5	57.6	44.3	65.3	70.7
avg-elmo	29.9	52.0	60.3	**47.5**	63.7	71.5
avg-avg	31.7	51.5	**60.5**	45.6	64.5	69.3

Table 5.2: Slovene-English word translation retrieval precision (in %) for **original forms** of words evaluated with different metrics. The first column represents the Slovene and English embedding type respectively. Abbreviation "fstx" represents the context-independent embedding with FastText. Abbreviations "flyr", "slyr", "avg", "elmo" refer to the first layer, second layer, average, and weighted sum of previous layers, respectively, of ELMo as described in Chapter 3. The columns nNN and nCSLS represent the evaluation methods described in Chapter 5 for different values of n.

Method / sl-en emb. type	1NN	5NN	10NN	1CSLS	5CSLS	10CSLS
fstx-fstx	**43.1**	**51.4**	52.3	43.1	48.1	51.4
flyr-flyr	13.9	33.3	41.9	29.1	49.1	56.8
flyr-slyr	10.4	30.4	40.5	25.1	50.9	58.4
flyr-elmo	16.3	36.3	42.4	28.5	52.5	58.7
flyr-avg	13.9	34.4	44.0	27.2	52.3	59.7
slyr-flyr	17.3	34.7	45.6	31.7	50.7	58.4
slyr-slyr	27.7	46.4	**54.1**	**43.5**	**65.3**	**72.5**
slyr-elmo	26.1	46.4	53.1	39.7	64.3	68.8
slyr-avg	26.1	47.2	53.6	41.1	63.7	69.6
avg-flyr	16.3	36.8	44.5	31.5	50.9	58.1
avg-slyr	22.9	45.1	52.8	42.4	64.0	68.5
avg-elmo	24.3	44.8	52.5	40.3	61.6	68.3
avg-avg	24.3	44.5	**54.1**	39.5	61.6	66.7

Table 5.3: Slovene-English word translation retrieval precision (in %) for **lemmas** of words evaluated with different metrics. The first column represents the Slovene and English embedding type respectively. Abbreviation "fstx" represents the context-independent embedding with FastText. Abbreviations "flyr", "slyr", "avg", "elmo" refer to the first layer, second layer, average, and weighted sum of previous layers, respectively, of ELMo as described in Chapter 3. The columns nNN and nCSLS represent the evaluation methods described in Chapter 5 for different values of n.

Method \ Size		500	1,000	5,000	10,000	20,000	35,000	50,000
tok slyr-slyr	1NN	8.3	16.5	31.5	36.0	37.1	40.3	**41.1**
	5NN	19.7	33.1	53.1	55.5	57.9	60.3	**60.8**
	10NN	28.3	41.9	60.3	62.4	65.6	**66.9**	66.7
	1CSLS	21.9	33.3	47.2	49.3	49.9	**51.2**	50.7
	5CSLS	39.5	54.1	66.4	66.4	**69.3**	68.8	69.1
	10CSLS	50.7	61.3	72.0	73.6	74.7	**74.9**	74.7
tok fstx-fstx	1NN	19.9	23.9	33.5	31.9	35.6	**36.3**	35.5
	5NN	25.1	31.1	41.0	42.2	45.4	46.2	**47.0**
	10NN	28.7	32.7	45.0	47.0	49.4	49.4	**51.0**
	1CSLS	20.3	25.1	32.7	32.3	34.7	**35.1**	34.7
	5CSLS	24.7	32.3	39.0	39.8	40.6	42.6	**43.4**
	10CSLS	26.7	34.7	42.2	44.6	43.4	47.4	**48.2**
lem fstx-fstx	1NN	27.3	34.3	43.1	43.5	45.8	**47.2**	46.8
	5NN	30.6	38.9	51.4	50.9	**53.7**	**53.7**	**53.7**
	10NN	33.3	41.7	52.3	54.6	**57.9**	55.6	56.5
	1CSLS	27.3	35.7	43.1	44.9	47.7	**50.0**	48.6
	5CSLS	31.5	40.3	48.1	53.7	54.2	56.0	**56.5**
	10CSLS	31.9	42.1	51.4	56.5	59.3	59.7	**60.2**

Table 5.4: Slovene-English word translation retrieval precision (in %) for various dataset sizes. The first column represents the token type and Slovene-English embedding type. Abbreviations "fstx" and "slyr" represent the context-independent embedding with FastText and the embedding of the second layer of ELMo, respectively. nNN and nCSLS represent the evaluation methods described in Chapter 5 for different values of n.

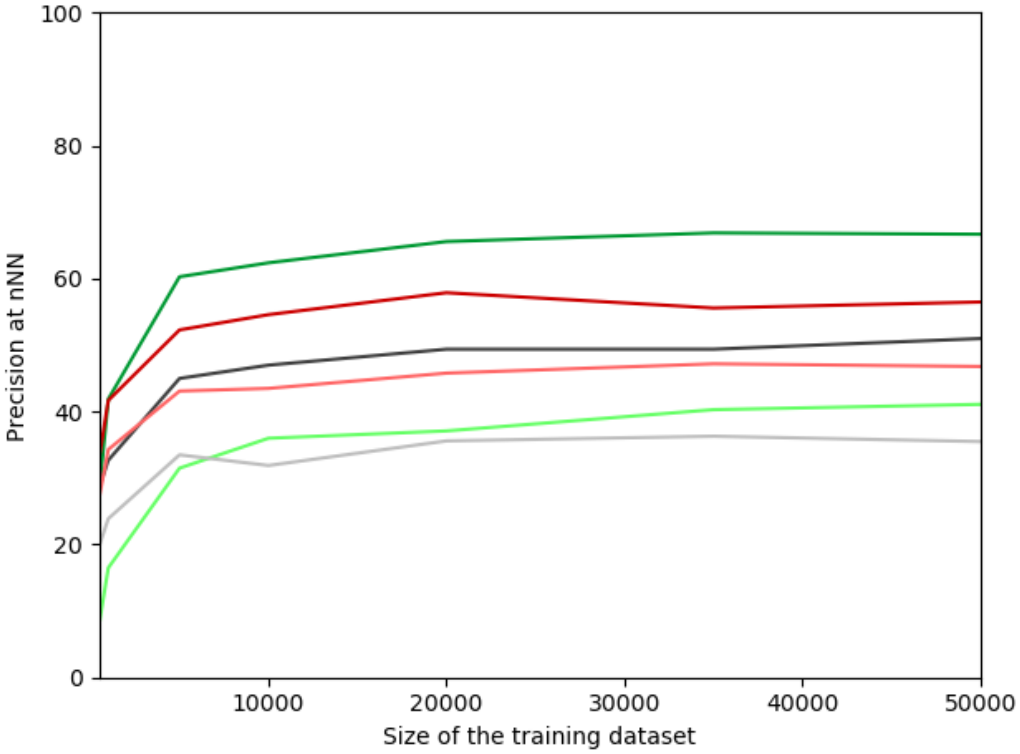

Figure 5.3: Word translation precision evaluated on different dataset sizes expressed with 1NN and 10NN measures. Light colors represent precision at 1NN and dark colors represent precision at 10NN. Black lines represent baselines – precision when aligning context-independent embeddings of the original form of words. Green lines represent the most precise method – precision when aligning the second layers of ELMo models. Red lines represent precision when aligning context-independent embeddings of lemmas.

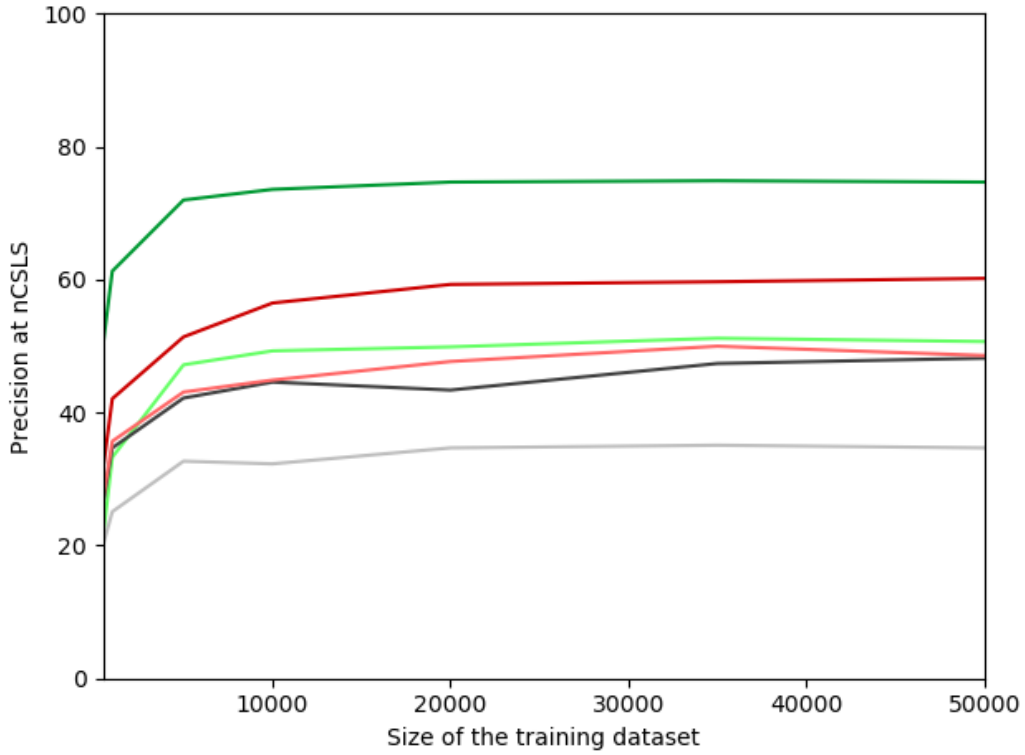

Figure 5.4: Word translation precision evaluated on different dataset sizes expressed with 1CSLS and 10CSLS measures. Light colors represent precision at 1NN and dark colors represent precision at 10CSLS. Black lines represent baselines – precision when aligning context-independent embeddings of the original form of words. Green lines represent the most precise method – precision when aligning the second layers of ELMo models. Red lines represent precision when aligning context-independent embeddings of lemmas.

Chapter 6

Conclusions

The main objective of our analysis was to propose a new method for cross-lingual embeddings that aligns contextual embeddings and improves the previous methods that use context-independent embeddings. The proposed method outperforms the baseline and performs reasonably well compared to methods proposed by other researchers, but evaluated on different languages.

We first obtained raw data that had many sentences translated into multiple languages. The proposed method was implemented for Slovene and English. The raw data was processed in such a way that we obtained a parallel corpus. Tokens in the parallel corpus were contextually embedded in the sentence in which they appeared. Two types of tokens were embedded: original word form and lemmas of words in the parallel corpus. Some of the entries in the parallel corpus were used for training the alignment procedure, others served for evaluation of the alignments on the word retrieval task.

We considered two main scenarios in our analysis. Tokens were contextually embedded with ELMo, which is an architecture that has multiple outputs. First, we were interested to know which of the possible mappings is the most precise; therefore evaluated several mappings. Alignment of the second layer of the Slovene and English ELMo models almost always outperformed others. Therefore, we have chosen it to assess how the size of

the training dataset affects precision. Unsurprisingly, the larger the training dataset is, the better the results are. Aligning original forms of words gives better results than lemmas. This is also intuitive since original forms contain more information about the semantic meanings of the tokens. There is a hypothesis that the proposed method performs significantly better than the alignment of context-independent embeddings, which we took as a baseline for the quality of the solution. We did not test the hypothesis of this analysis, but it is an idea for future work. When compared to the results of similar methods for other languages, we obtained slightly worse results. The reasons for that could lead to further improvement of our method.

A strength of the proposed method is that it does not require many bilingual resources. With only 20,000 training entries we reached satisfactory results. If each of the tokens appears in a different sentence, we need at most 20,000 sentences, which is available for many language pairs. The weakness of our method is that it depends on the architecture for contextual embeddings. First, the contextual embedding of tokens is time-consuming. The ELMo contextual embedding uses LSTM, which requires the output of one token as input for the following token. Because of that, they cannot be easily parallelized. However, this issue can be solved with better architectures for contextual embeddings. Furthermore, it can be mitigated by embedding the tokens only in the sentence in which they appear and embedding these sentences using multiple processors. Second, a model for contextual embeddings is required and training it requires lots of time. However, we can almost always find pre-trained models, as we did for our analyses. Moreover, it can also be improved with new architectures that would require less time for training.

We used a supervised method in the alignment procedure. Although it required a small bilingual parallel corpus to produce satisfactory results, for some languages it is better if we can solve the problem in an unsupervised manner. Researchers have already found several such successful methods.

However, none of them has tried to do it by aligning centroids of the source and target spaces. As further work, the idea is to find a few anchor points without supervision. For instance, centroids of the vectors in the source and target languages should be aligned. The furthest points from the centroids may also be aligned. In that way, we can remove the furthest points and then continue aligning the next furthest points and so on for a desired number of iterations. Implementing a successful unsupervised alignment of embeddings would mean that we do not require bilingual resources at all. This would be very useful for language pairs with limited resources.

Bibliography

[1] CLARIN.SI. `https://github.com/clarinsi`. Accessed: 25.06.2019.

[2] Hanan Aldarmaki and Mona Diab. Context-Aware Crosslingual Mapping. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[3] 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.

[4] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, 2017.

[5] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, 2018.

[6] R. Bellman, Rand Corporation, and Karreman Mathematics Research Collection. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957.

[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[8] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word Translation Without Parallel Data. In *International Conference on Learning Representations (ICLR)*, 2017.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018.

[10] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[11] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[12] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[14] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations (ICLR)*, 2017.

[15] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[16] Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc., 2014.

[17] Nikola Ljubešić and Tomaž Erjavec. Corpus vs. Lexicon Supervision in Morphosyntactic Tagging: the Case of Slovene. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, 2016. European Language Resources Association (ELRA).

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR) Workshop Papers*, 2013.

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

[20] 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.

[21] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contex-

tualized word representations. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

[22] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[23] Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep):2487–2531, 2010.

[24] Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, Mar 1966.

[25] Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1599–1613, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[26] Ralf Steinberger, Mohamed Ebrahim, Alexandros Poulis, Manuel Carrasco-Benitez, Patrick Schlüter, Marek Przybyszewski, and Signe Gilbro. An overview of the European Union's highly multilingual parallel corpora. *Language resources and evaluation*, 48(4):679–707, 2014.

[27] Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. DGT-TM: A freely available translation memory in 22 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 454–459,

Istanbul, Turkey, May 2012. European Languages Resources Association (ELRA).

[28] Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA).

[29] Wilson L. Taylor. "Cloze Procedure": A New Tool for Measuring Readability. *Journalism Bulletin*, 30(4):415–433, 1953.

[30] Jörg Tiedemann. DGT. `http://opus.nlpl.eu/DGT-v4.php`. Accessed: 24.06.2019.

[31] Jörg Tiedemann. Parallel Data, Tools and Interfaces in OPUS. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[33] 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.

[34] Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1970, Vancouver, Canada, July 2017. Association for Computational Linguistics.