EMBEDDIA

Cross-Lingual Embeddings for Less-Represented Languages in European News Media

Research and Innovation Action
Call: H2020-ICT-2018-1
Call topic: ICT-29-2018 A multilingual Next generation Internet
Project start: 1 January 2019 Project duration: 36 months

D1.2: Initial cross-lingual and multilingual embeddings technology (T1.1)

Executive summary

Text embedding has become an established language representation used in natural language processing with deep neural networks. Cross-lingual embeddings are the means to establish maps between embeddings for different languages and enable transfer of trained model from resource-rich to less-resourced languages. This deliverable presents the most important word embeddings and cross-lingual approaches, followed by the analysis of existing cross-lingual embedding approaches that map monolingual embeddings. On three cross-lingual evaluation tasks, i.e. dictionary induction, word analogy, and named entity recognition, we demonstrate that the vecmap approach (supervised or unsupervised) is superior to the MUSE approach for both non-contextual and contextual embeddings. Recent developments in the area of cross-lingual alignments show that our work on non-isomorphic cross-lingual alignments is promising.

Partner in charge: UL

Project co-funded by the European Commission within Horizon 2020 **Dissemination Level**		
PU	Public	PU
PP	Restricted to other programme participants (including the Commission Services)	–
RE	Restricted to a group specified by the Consortium (including the Commission Services)	–
CO	Confidential, only for members of the Consortium (including the Commission Services)	–

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825153

Deliverable Information

Document administrative information	
Project acronym:	**EMBEDDIA**
Project number:	**825153**
Deliverable number:	**D1.2**
Deliverable full title:	**Initial cross-lingual and multilingual embeddings technology**
Deliverable short title:	**Initial cross-lingual embeddings**
Document identifier:	**EMBEDDIA-D12-InitialCrosslingualEmbeddings-T11-submitted**
Lead partner short name:	**UL**
Report version:	**submitted**
Report submission date:	**31/12/2019**
Dissemination level:	**PU**
Nature:	**R = Report**
Lead author(s):	**Marko Robnik-Šikonja (UL), Matej Ulčar (UL), Gregor Jerše (UL)**
Co-author(s):	**Matthew Purver (QMUL), Marko Pranjić (Trikoder), Senja Pollak (JSI), Blaž Škrlj (JSI)**
Status:	**_ draft, _ final, x submitted**

The EMBEDDIA Consortium partner responsible for this deliverable has addressed all comments received. Changes to this document are detailed in the change log table below.

Change log

Date	Version number	Author/Editor	Summary of changes made
01/10/2019	v1.0	Marko Robnik-Šikonja (UL)	First draft.
10/10/2019	v1.1	Gregor Jerše (UL)	Some contents about mappings.
05/11/2019	v1.2	Matej Ulčar (UL)	Added evaluation results.
06/11/2019	v1.3	Marko Robnik-Šikonja (UL)	Checked and consolidated the report.
20/11/2019	v1.4	Matthew Purver (QMUL)	Updated template, added outputs section.
01/12/2019	v1.5	Senja Pollak (JSI)	Added language comparison paper and internal review.
02/12/2019	v1.6	Antoine Ducet (ULR)	Internal review.
07/12/2019	v1.7	Marko Robnik-Šikonja (UL), Matej Ulčar (UL)	Revision based on internal reviews.
08/12/2019	v1.8	Nada Lavrač (JSI)	Report quality checked and finalised.
09/12/2019	final	Nada Lavrač (JSI), Marko Robnik-Šikonja (UL)	Revision based on quality check.
23/12/2019	submitted	Tina Anžič (JSI)	Report submitted.

Table of Contents

List of abbreviations

NLP	Natural Language Processing
NER	Named Entity Recognition
LSTM	Long Short-term Memory
CSLS	Cross-domain Similarity Local Scaling
CNN	Convolutional Neural Network
ELMo	Embeddings from Language Models
BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag Of Words
MLM	Masked Language Model

1 Introduction

The EMBEDDIA project aims to improve cross-lingual transfer of language resources and trained models using word embeddings and cross-lingual word embeddings. EMBEDDIA works with *nine languages*: English, Slovene, Croatian, Estonian, Lithuanian, Latvian, Russian, Finnish, and Swedish. We presented the basic description of embeddings and cross-lingual embeddings in Deliverable *D1.1 Datasets, benchmarks and evaluation metrics for cross-lingual word embeddings*. To make this document self-contained, we first repeat some basic explanations in Section 1.1, which a reader acquainted with embeddings can skip. Section 1.2 outlines the context of this deliverable within the EMBEDDIA project and presents the structure of this report.

1.1 Introducing embeddings

To process text, neural networks require numerical representation of the given text (words, sentences, documents), referred to as **text embeddings**. In this report we focus on word embeddings, as the main ingredient of text embeddings.

Word embeddings are representations of words in numerical form, as vectors of typically several hundred dimensions. The vectors are used as an input to machine learning models; for complex language processing tasks these are typically deep neural networks. The embedding vectors are obtained from specialized neural network-based embedding alsogirhms, e.g., word2vec (Mikolov, Le, & Sutskever, 2013), GloVe (Pennington et al., 2014), or fastText (Bojanowski et al., 2017). For training, the embedding algorithms use large monolingual text collections (called corpora) that encode important information about word meaning as distances between the embedded vectors. To enable downstream machine learning on text understanding tasks, the embeddings shall retain semantic relations between words, which should be preserved even across languages.

Currently, the best known word embeddings are produced by the word2vec method (Mikolov, Sutskever, et al., 2013), which we use as a baseline in this report. The problem with word2vec embeddings is their failure to express polysemous words. During training of an embedding, all senses of a given word (e.g., *paper* as a material, as a newspaper, as a scientific work, and as an exam) contribute relevant information in proportion to their frequency in the training corpus. This causes the final vector to be placed somewhere in the weighted middle of all the word's meanings. Consequently, rare meanings of words are poorly expressed with word2vec and the resulting vectors do not offer good semantic representations. For example, none of the 50 closest vectors of the word *paper* is related to science[1]. This problem is addressed by **contextual embeddings**, where the idea is to generate a different vector for each context a word appears in, with the notion of context typically defined as the surrounding sentence. To a large extent, this solves the problems with word polysemy, i.e. the context of a sentence is typically enough to disambiguate different meanings of a word for humans and so it is for the learning algorithms. In our work we mostly use, analyze, and improve upon currently the most successful approaches to contextual word embeddings, ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019).

The state-of-the-art in embeddings is rapidly progressing. Modern word embedding spaces exhibit similar structures across languages, even when considering distant language pairs like English and Vietnamese (Mikolov, Le, & Sutskever, 2013). This means that embeddings independently produced from monolingual text resources can be aligned (Mikolov, Le, & Sutskever, 2013), resulting in a common cross-lingual representation, called **cross-lingual embedding**, which allows for fast and effective integration of information in different languages.

[1] A demo showing near vectors computed with word2vec from Google News corpus is available at `http://bionlp-www.utu.fi/wv_demo/`.

1.2 Objectives and structure of the report

The objectives of workpackage WP1 of the EMBEDDIA project are to advance cross-lingual and context-dependent word embeddings and test them with deep neural networks. This report describes the results of the work performed in T1.1 in the first 12 months of project duration. The specific objective of T1.1 is to advance cross-lingual and multilingual word embeddings technology. The main contributions presented in this report (in the order of appearance) are as follows:

1. development of a dataset and a novel alignment approach for contextual embeddings (e.g., ELMo) based on a dictionary and parallel corpus, described in Section 3.1;
2. initial development of a novel method for cross-lingual alignment of non-contextual and contextual embeddings based on locally isomorphic transformations, presented in Section 3.2;
3. evaluation of the main existing cross-lingual mapping techniques using both intrinsic and extrinsic tasks, described in Section 4;
4. development of publicly available culture-independent monolingual analogy datasets for all EMBEDDIA languages, and culture-independent cross-lingual analogy datasets for all combinations of EMBEDDIA languages, described in Section 4.2 and in the appended paper by Ulčar & Robnik-Šikonja (2019b), submitted to the LREC-2020 conference;
5. development of a novel network topology based approach for language comparison that may be useful in unsupervised alignment, described in Section 5 and in the appended paper by Škrlj & Pollak (2019), published in Proceedings of the International Conference on Statistical Language and Speech Processing 2019.

The above objectives and contributions are slightly different from the ones anticipated in the EMBEDDIA project proposal: at the time of proposal writing we namely anticipated that contextual embedding methods will need to be developed from scratch. However, due to recently developed highly successful contextual embedding approaches ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), part of the original EMBEDDIA objectives were already successfully achieved by AllenNLP and Google research groups, which have huge amounts of resources available (AllenNLP developed ELMo and Google developed BERT). While (according to the definitions in the EMBEDDIA proposal) ELMo and BERT are context-dependent and dynamic, and as such address the objectives of Task 1.2 of EMBEDDIA, we describe them in this deliverable (D1.2 addressing Task 1.1), given that they represent state-of-the-art techniques also for cross-lingual and multilingual embedding technologies. Specifically, a variant of BERT, called multilingual BERT, was trained on 104 languages and works well in cross-lingual setting with no need for any further alignments.[2] Moreover, both ELMo and BERT use subword input, which is very appropriate for morphologically rich languages (addressing also some of the objectives of Task 1.3). Given the described developments, we have therefore in Task 1.1 re-focused our research on the challenging remaining issues of cross-lingual alignment for contextual embeddings rather than on non-contextual embeddings as originally planned.

This report is split into eight further sections. In Section 2, we describe non-contextual and contextual embeddings used, followed by the description of existing cross-lingual embeddings in Section 2.3. As recently contextual embeddings have turned out to be much more successful compared to non-contextual embeddings, we propose two new approaches for alignment of these embeddings in Section 3. We present the evaluation scenarios and the results of experimental evaluation of cross-lingual mappings in Section 4. Section 5 presents an approach to detecting language similarity, potentially useful to predict the success of unsupervised cross-lingual embeddings by focusing their application to most similar languages. Availability of new resources produced in this work is discussed in Section 6. Section 7 summarises our conclusions regarding cross-lingual mappings between embeddings and outlines the plans for further work. The two appendices include the papers by Ulčar & Robnik-Šikonja (2019b) and by Škrlj & Pollak (2019).

[2] For BERT, which uses a novel non-recurrent deep architecture, called Transformer, extraction of explicit numeric vectors (i.e. embeddings) is questionable, and the model is mostly used as a whole, with only the last layer removed and retrained.

2 Background: Text embedding models and cross-lingual embeddings

Historically, text embeddings started with sparse representation of documents in the bag-of-words form and latent semantic analysis, i.e. matrix decomposition of term-term matrices Landauer et al. (1998). The resulting word vectors are linear combinations of original words, but the linearity was soon sacrificed for better preservation of semantic properties with neural network-based embeddings which we present in Section 2.1. We briefly outline the most popular word2vec (Mikolov, Le, & Sutskever, 2013) embedding approach. As these embeddings have problems with ambiguous words, recent developments take the context of words into account, as discussed in Section 2.2. We present the ELMo and BERT embeddings, which are based on deep neural language models.

2.1 Non-contextual embeddings

As deep neural networks became the predominant learning method for text analytics, it was quite natural that they also gradually became the method of choice for text embeddings. A procedure, common to these embeddings, is to train a neural network on one or more semantic text classification tasks and then take the weights of the trained neural network as a representation for each text unit (word, n-gram, sentence, or document).

The labels required for training such a classifier come from huge corpora of available texts. Typically, they reflect word co-occurrence, like prediction of the next or previous word in a sequence, and filling in missing words, but may be extended with other related tasks, such as sentence entailment. The positive instances for training are obtained from the text in the used corpora, while the negative instances are mostly obtained with negative sampling (sampling from instances that are highly unlikely related).

As an example of non-contextual embedding methods, we present word2vec method in Section 2.1.1, which trains a shallow neural network and produces a single vector for each word. In Section 2.2, we continue with contextual word embeddings which train deeper networks and combine weights from different layers to produce unique vectors for each word occurrence, based on the sentence it appears in. These contextual word embeddings are the current state of the art in text representation.

2.1.1 Word2vec embedding

Mikolov, Sutskever, et al. (2013) introduced the word2vec method, and made it immensely popular by training it on a huge Google News data set (about 100 billion words), and making the pretrained 300-dimensional vectors for 3 million English words and phrases publicly available.[3] Word2vec consists of two related methods, *continuous bag of words (CBOW)* and *skip-gram*. Both methods construct a neural network for classification of co-occurring words by taking a word and its d preceding and succeeding words, e.g., ± 5 words.

CBOW takes the neighboring words and predicts the central word. Conversely, the skip-gram variant takes the word and predicts its neighborhood. The actual neural network is similar in both cases, i.e. there is one word on the input (either the neighboring word for CBOW or central word for skip-gram method) and one word on the output, both represented in one-hot encoding. Empirical evaluations have shown a slight advantage of skip-gram model over CBOW for many tasks, therefore we focus on it in the reminder of this section.

The words and their contexts (one word at a time) appearing in the training corpus constitute the training instances of the classification problem. Assuming that we have the context window of size $d = 2$ and the sentence *Jana is watching her linear algebra lecture with new glasses*, we generate the following positive instances for the word *linear*:

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

(linear, watching), (linear, her), (linear, algebra), (linear, lecture).

The first word of the training pair is presented at the input of the network in 1-hot-encoding representation. The network is trained to predict the second word. The difference in prediction is evaluated using the criterion function. For a sequence of T training words $w_1, w_2, w_3, \dots, w_T$, the skip-gram model maximizes the average log probability of

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

To make the process efficient for the 100 billion Google News corpus, the actual implementation uses several approximation tricks. The biggest problem is the estimation of $p(w_{t+j}|w_t)$ which normally requires a computation of dot product between w_t and all other words in the vocabulary. Mikolov, Sutskever, et al. (2013) solve this issue with negative sampling, which replaces $\log p(w_{t+j}|w_t)$ terms in objective function with the results of logistic regression classifiers trained to distinguish between similar and dissimilar words.

Once the network is trained we can produce the vectors for each word in the vocabulary. One-hot-encoding of a word only activates one input connection for each hidden layer neuron. The weights on these connections constitute the embedding vector for the given input word. In principle, we could return the weights between the input and the hidden layer or between the hidden layer and the output layer, but mostly the former weights are used.

The properties of the resulting word embeddings depend on the size of the context. For lower number of neighboring words, we get embeddings that perform better on syntactic tasks (e.g., $\pm$ 5 words). For larger neighborhoods (e.g., $\pm$ 10 words) the embeddings better express semantic properties. There is also some difference between the CBOW and skip-gram variant. CBOW performs well for syntactic problems, while skip-gram variant works comparably well for syntactic problems as CBOW, and better for semantic problems.

2.1.2 fastText embedding

Bojanowski et al. (2017) developed the fastText method, built upon the word2vec method but introduced a subword information, which is more appropriate for morphologically rich languages such as the ones processed in EMBEDDIA. They took skip-gram method from word2vec and edited the scoring function used to calculate the probabilities (see Section 2.1.1). In the word2vec method, this scoring function is equal to a dot product between two word vectors. For words w_t and w_c and their respective vectors u_t and u_c, the scoring function s is equal to $s(w_t, w_c) = \mathbf{u}_t^\top \mathbf{u}_c$. The scoring function in fastText is a sum of dot products for each subword (i.e. character n-gram) that appears in the word w_t:

$$s(w_t, w_c) = \sum_{g\in G_t} \mathbf{z}_g^\top \mathbf{u_c},$$

where $\mathbf{z}_g$ is a vector representation of an n-gram (subword) g and G_t is a set of all n-grams (subwords) appearing in w_t. As fastText is conceptually very similar to word2vec, we do not treat them as different methods, but only test fastText.

2.2 Contextual embeddings

With word2vec embedding, the text mining community gained a powerful tool and soon the word2vec precomputed embeddings became a popular choice for the first layers of most classification deep neural networks. The problem with word2vec embeddings is their failure to express polysemous words. Contextual word embeddings generate a different vector for each context a word appears in and the context is typically defined sentence-wise. This solves the problems with word polysemy to a large extent.

In this section, we describe different approaches to take word context into account used in modern embeddings. There are two baseline premises common to them, the concepts of language model and transfer learning. We first describe language models below, while transfer learning approaches are discussed together with the two most successful implementations to contextual word embeddings, ELMo (Peters et al., 2018) in Section 2.2.1 and BERT (Devlin et al., 2019) in Section 2.2.2.

Language model (LM) is a probabilistic prediction model that learns a distribution of words in a given language. For example, for a sequence of words $w_1 w_2 \dots w_n$, a language model returns its probability $p(w_1 w_2 \dots w_n)$. Frequently, language models are used to predict the next word in a sequence, i.e. $p(w_{n+1}|w_1 w_2 \dots w_n)$. Traditionally, language models were trained on large textual corpora using n-grams, assuming independence of words beyond certain distance. For example, if we assume that a word only depends on the previous one in the sequence $p(w_{n+1}|w_1 w_2 \dots w_n) = p(w_{n+1}|w_n)$, we can only count all joint occurrences of pairs of words. Let $c(w_i)$ denote a count of word w_i in a corpus and $c(w_i w_j)$ a count of joint occurrence of words w_i and w_j. To compute the probability of the next word using bigrams we then use

$$p(w_{n+1}|w_n) = \frac{c(w_n w_{n+1})}{c(w_n)}.$$

The probability of a sequence is retrieved using Bayes chain rule for conditional probabilities

$$p(w_1, w_2, \dots w_n) = p(w_1)p(w_2|w_1) \dots p(w_n|w_1 w_2 \dots w_{n-1}).$$

Using Markov independence assumption for sequences longer than 2 words, we get

$$p(w_1, w_2, \dots w_n) = p(w_1)p(w_2|w_1) \dots p(w_n|w_{n-1}).$$

The remaining conditional probabilities can be estimated using bigram and unigram counts as above.

In real texts, Markov independence assumption is not true even for very long sequences (think of split verbs or dependent sentences separating a noun and its verb). The n-gram counting method therefore does not work well. Additionally, frequencies of n-grams for $n > 3$ become statistically very unreliable even with huge corpora.

Lately, language models are trained using deep neural networks. If a neural network is trained to predict the next word in a sequence from a large text corpus, the sequences can actually be much longer and we still get reliable results. Language models can also be trained in the reverse direction, i.e. for a backwards language model ($\overleftarrow{LM}$) we train a network to predict $p(w_i|w_{i+1} w_{i+2} \dots w_{i+k})$. Further generalization of LMs are called masked language models (MLMs) which predict a missing word anywhere in a sequence, mimicking a gap filling cloze test $p(w_i|w_{i-b} w_{i-b+1} \dots w_{i-1} w_{i+1} \dots w_{i+f})$. For n-gram approach this would be unfeasible but neural networks are flexible enough and can successfully predict the gaps.

2.2.1 ELMo

ELMo (Embeddings from Language Models) embedding (Peters et al., 2018) is an example of a state-of-the-art pre-trained transfer learning model. The first layer is a CNN layer, which operates on a character level. It is context independent, so each word always gets the same embedding, regardless of its context. It is followed by two biLM (bidirectional language model) layers. A biLM layer consists of two concatenated LSTMs. In the first LSTM, we try to predict the following word, based on the given past words, where each word is represented by the embeddings from the CNN layer. In the second LSTM, we try to predict the preceding word, based on the given following words. It is equivalent to the first LSTM, just reading the text in reverse.

The actual embeddings are constructed from the internal states of a bidirectional LSTM neural network. Higher-level layers capture context-dependent aspects, while lower-level layers capture aspects of syntax (Peters et al., 2018). To train the ELMo network, we put one sentence at a time on the input and the representation of each word depends on the whole sentence, i.e. it reflects the contextual features of the input text and thereby polysemy of words. For an explicit word representation, one can use only the top layer but more frequently one combines all layers into a vector. The representation of a word or a token t_k at position k is composed from

$$R_k = \{x_k^{LM}, \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} \mid j = 1, \ldots, L\} \tag{1}$$

where L is the number of layers (ELMo uses $L = 2$), index j refers to level of bidirectional LSTM network, x is the initial token representation (either word or character embedding), and h^{LM} denotes hidden layers of forward or backward language model.

In NLP tasks, any set of these embeddings may be used, however, a weighted average is usually used. The weights of the average are learned during the training of the model for the specific task. Additionally, an entire ELMo model can be fine-tuned on a specific end task.

At the time of its introduction, ELMo has been shown to outperform previous pre-trained word embeddings like word2vec and GloVe on many NLP tasks, e.g., question answering, named entity extraction, sentiment analysis, textual entailment, semantic role labeling, and coreference resolution (Peters et al., 2018).

We trained high-quality ELMo embedding on large corpora for all EMBEDDIA languages (Ulčar & Robnik-Šikonja, 2019a). Details of the training and properties of the obtained models are described in *D1.3 Initial context-dependent and dynamic embeddings technology*.

2.2.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) embedding (Devlin et al., 2019) generalises the idea of language models to masked language models, inspired by the cloze test, which tests understanding of a text by removing certain portion of words, which the participant is asked to replace. The masked language model randomly masks some of the tokens from the input, and the task of LM is to predict the missing token based on its neighbourhood. BERT uses transformer architecture of neural networks (Vaswani et al., 2017) in a bidirectional sense and further introduces the task of predicting whether two sentences appear in a sequence. The input representation of BERT are sequences of tokens representing subword units. The input is constructed by summing the embeddings of corresponding tokens, segments, and positions. Some very common words are kept as single tokens, others are split into subwords (e.g., common stems, prefixes, suffixes—if needed down to a single letter tokens). The original BERT project offers pre-trained English, Chinese and multilingual models. The latter is trained on 104 languages simultaneously and covers all EMBEDDIA languages.

To use BERT in classification tasks only requires adding connections between its last hidden layer and new neurons corresponding to the number of classes in the intended task. The fine-tuning process is applied to the whole network and all of the parameters of BERT and new class specific weights are fine-tuned jointly to maximize the log-probability of the correct labels.

BERT has shown excellent performance on 11 NLP tasks: 8 from GLUE language understanding benchmark (Wang et al., 2018), question answering, named entity recognition, and common-sense inference Devlin et al. (2019). The performance on monolingual tasks has often improved upon ELMo. However, while multilingual BERT covers 104 languages, its subword dictionary is composed of tokens for all involved languages, which might not be optimal for a particular language. Further, similarly to ELMo, its training and tuning are computationally highly demanding tasks, out of reach for most researchers.

We discuss issues on training ELMo and BERT models for EMBEDDIA languages in deliverable *D1.3 Initial context-dependent and dynamic embeddings technology*.

2.3 Cross-lingual embeddings

Word embeddings represent each word in a language as a vector in a high dimensional vector space so that the relations between words in a language are reflected in their corresponding embeddings. Cross-lingual embeddings attempt to map words represented as vectors from one vector space to the other, so that the vectors representing words with the same meaning in both languages are as close as possible. Søgaard et al. (2019) present a detailed overview and classification of cross-lingual methods.

We present approaches to find cross-lingual mappings sorted into two groups. The first group of approaches, presented in Section 2.3.1, uses monolingual embeddings with the optional help from bilingual dictionary to align the embeddings. The second group of approaches, presented in Section 2.3.2, uses bilingually aligned (comparable or even parallel) corpora for joint construction of embeddings in all involved languages.

In this report, we empirically evaluate only the methods from the monolingual mapping approaches (Section 4) The second group of methods projecting into a joint space is recently dominated by the contextual embedding methods, e.g., BERT (Devlin et al., 2019), described in Section 2.2.2. Pretrained multilingual BERT is typically used as starting model to be fine-tuned for a particular task, without explicitly extracting embedding vectors. We report the evaluation results of BERT used directly on downstream tasks in several languages in *D1.4 Initial deep network architecture*.

2.3.1 Alignment of monolingual embeddings

Cross-lingual alignment methods take precomputed word embeddings for each language and align them with an optional use of bilingual dictionaries. Two types of monolingual embedding alignment methods exist. The first type of approaches map vectors representing words in one of the languages into the vector space of the other language (and vice-versa). The second type of approaches maps embeddings from both languages into a common vector space. The goal of both types of alignments is the same: the embeddings for words with the same meaning must be as close as possible in the final vector space. A comprehensive summary of existing approaches can be found in (Artetxe et al., 2018a).

The open source implementation of the method described in Artetxe et al. (2018b,a), named *vecmap*[4], is able to align monolingual embeddings either using supervised, semi-supervised or unsupervised approach.

The supervised approach requires the use of a large bilingual dictionary, which is used to match embeddings of same words. Then embeddings are aligned using the Moore-Penrose pseudo-inverse which minimizes the sum of squared Euclidean distances. The algorithm always converges but can be caught in a local maximum when the initial solution is poor. To overcome this, several methods (stochastic dictionary introduction, frequency-based vocabulary cutoff, etc) are used that help the algorithm to climb out of local maximums. A more detailed description of the algorithm is given in (Artetxe et al., 2018b).

The semi-supervised approach uses a small initial seeding dictionary, while the unsupervised approach is run without any bilingual information. The latter uses similarity matrices of both embeddings to build an initial dictionary. This initial dictionary is usually of poor but sufficient quality for later processing. After the initial dictionary (either by seeding dictionary or using similarity matrices) is built, the iterative algorithm is applied. The algorithm first computes optimal mapping using pseudo-inverse approach for the given initial dictionary. Then optimal dictionary for the given embeddings is computed and the procedure is repeated with the new dictionary.

When constructing mappings between embedding spaces, a bilingual dictionary can be helpful as its entries can be used as anchors for the alignment map for supervised and semi-supervised approaches.

[4] `https://github.com/artetxem/vecmap`

However, lately researchers have proposed approaches that do not require the use of bilingual dictionary, but rely on adversarial approach (Conneau et al., 2018) or use the frequencies of the words (Artetxe et al., 2018b) in order to find a required transformation. These are called unsupervised approaches.

The Facebook research project MUSE[5] can find a cross-lingual map with the use of bilingual dictionary (supervised) or without one (unsupervised approach). Unsupervised approach works by using adversarial training to find the starting linear mapping. From this mapping, a synthetic dictionary is extracted, which is used to fine-tune the starting mapping using Procrustes approach, described in detail by Conneau et al. (2018).

2.3.2 Projecting into a common vector space

To construct a common vector space for all involved languages, we require a large aligned bilingual or multilingual parallel corpus. The constructed embeddings must map the same words in different languages as close as possible in the common vector space. The availability and quality of alignments in training set corpus may present an obstacle. While Wikipedia, subtitles, and translation memories are good sources of aligned texts for large languages, less-resourced languages are not well-presented and building embeddings for such languages is a challenge.

LASER[6] (Language-Agnostic SEntence Representations) is a Facebook research project focusing on joint sentence representation for many languages (Artetxe & Schwenk, 2019). Similarly to machine translation architectures, it uses an encoder-decoder architecture. The encoder is trained on a large parallel corpora, translating a sentence in any language or script to a parallel sentence in either English or Spanish (whichever exists in the parallel corpus), thereby forming a joint representation of entire sentences in many languages in a shared vector space. The project focused on scaling to a large number of languages, currently the encoder supports 93 different languages, including all EMBEDDIA languages. The resulting joint embedding can be transformed back into a sentence using decoder for the specific language. This allows training a classifier working on data from just one language and use it on any language supported by LASER.

Multilingual BERT embedding (Devlin et al., 2019) also projects many languages into a joint space. Its description is contained in Section 2.2.2.

3 New contextual cross-lingual mapping approaches

Context-dependent models calculate a word embedding for each occurrence of a word, thus a word gets a different vector for each context. Mapping such vector spaces from different languages is not straightforward. Schuster et al. (2019) observed that vectors representing different occurrences of each word form clusters. They averaged the vectors for each word occurrence so that each word was represented with only one vector, a so called anchor. They applied the same procedure to both languages and aligned the anchors using supervised or unsupervised method of MUSE (Conneau et al., 2018). This method, however, comes with a loss of information. Many words have multiple meanings, which can not be simply averaged. For example, the word »mouse« can mean a small rodent or a computer input device. Context-dependent models correctly assign significantly different vectors to these two meanings, since they tend to appear in very different contexts. Further, a word in one language can be represented with several different words (one for each meaning) in another language, or vice versa. By averaging the contextual embedding vectors, we lose these distinctions in meaning. We propose two new methods which take different contexts and word meanings into account. The first approach uses parallel corpus and bilingual dictionary, but still uses the monolingual embedding mapping methods (described in Section 2.3.1) for alignment of contextual embeddings. The second approach uses the

[5] https://github.com/facebookresearch/MUSE
[6] https://github.com/facebookresearch/LASER

same resources but drops the assumption that the aligned spaces are isomorphic. These two methods are still work in progress and have not been satisfactorily evaluated in this report.

3.1 Contextual mapping with parallel corpus and dictionary assuming isomorphic spaces

We propose a novel method for alignment of contextual embeddings based on the idea of matching contexts in different languages. For that we require two resources, a sentence aligned parallel corpus of the two involved languages and their bilingual dictionary. The dictionary alone is not sufficient, as the words are not given in the context, therefore it cannot help for alignment of contextual embeddings. The parallel corpus alone is also not sufficient as the alignment is on the level of paragraphs or sentences, and not on the level of words. By combining both resources, we take a translation pair from the dictionary and find sentences in the parallel corpus, with one word from the pair present in the sentence of the first language and the second word from the translation pair present in the second language sentence. As a result we get matching words in matching contexts (sentences). With large enough collection of words in matching contexts, we compute their contextual embedding vectors using ELMo and align them with any of the non-contextual mapping methods, e.g., vecmap library (Artetxe et al., 2018a).

Our initial test of the described approach for cross-lingual mapping uses English and Slovene languages. We used a parallel corpus of EU translation memories (Tiedemann, 2012) from OPUS web page[7], in addition to a large list of translation pairs from Oxford English-Slovene bilingual dictionary with about 300,000 entries. We checked each word translation pair from our collection in the parallel corpus and collected sentences where one word exists in the first language sentence and its pair exists in the second language sentence. When such a match was found, the two words and their ELMo embeddings, computed on the matching contexts (sentences), were added to the list of anchors. This list was used to map one vector space to another, allowing us to map one word with multiple meanings in one language to multiple different words in another language.

We used the computed bilingually aligned contextual embedding pairs as an input to previously described methods that align two monolingual embeddings (Section 2.3.1). To get the cross-lingual alignment we used the vecmap method (Artetxe et al., 2018a).

Recently, a similar approach was proposed by Aldarmaki & Diab (2019) but did not use a high-quality dictionary as we did. Instead, they extracted a dictionary of contextualized words from the parallel corpora by first applying word-level alignments using Fast Align approach (Dyer et al., 2013). They then calculated the ELMo contextual embeddings for both aligned sentences, and extracted a dictionary from the aligned words that have a one-to-one alignment (i.e. they excluded phrasal alignments). Aldarmaki & Diab (2019) tested their approach only on similar languages (English, German, Spanish) and showed good results in sentence translation retrieval task, where they measured the accuracy of retrieving the correct translation from the target side of a test parallel corpus using nearest neighbor search and cosine similarity. Our intention is to use the above proposed approach and datasets on morphologically rich languages, to test a wide range of dictionary sizes and qualities, from small automatically extracted dictionaries to large professional dictionaries.

3.2 Locally isomorphic mapping

As several researchers have observed, the monolingual embedding spaces of two different languages are not completely isomorphic, which is especially true for distant languages (Ormazabal et al., 2019). This causes error in methods which assume isomorphism of embedding spaces, including vecmap and MUSE.

[7] http://opus.nlpl.eu/

We propose a new alignment method that assumes that monolingual embedding spaces are locally isomorphic. This leads to the following modifications:

- Using the same dataset of bilingually aligned contextual embedding pairs as above (Section 3.1), we modify the vecmap, which originally computes a single linear mapping between the entire embedding spaces. We choose a set of words with their contextual embedding vectors in the first monolingual space and compute the Voronoi cells of their vectors in this embedding space. The Voronoi cell of a given word vector consists of all points in the embedding space closer to this vector than to vectors of any other word in the set.
- The words are chosen in such a way that there is approximately the same number of words in each Voronoi cell. In the next step, we compute the alignment map separately for each Voronoi cell using the vecmap algorithm in supervised mode, thus producing one linear map for every Voronoi cell. The maps of all Voronoi cells are merged and represent a joint mappings for the entire embedding space. As the mappings between Voronoi cells are different the resulting mapping is not isomorphic but only locally isomorphic, which shall improve the properties of the resulting embedding.

The method and its empirical evaluation will be described in deliverable D1.6 at M24, as at the time of writing this report the method has not been fully developed and tested to be included in this deliverable.

4 Cross-lingual evaluation scenarios and results

We evaluated the produced cross-lingual embeddings on several intrinsic and extrinsic tasks. A detailed description of these tasks, including the datasets and evaluation metrics is part of deliverable *D1.1 Datasets, benchmarks and evaluation metrics for cross-lingual word embeddings*. Here, we shortly explain their role in cross-lingual mapping scenarios.

As baseline context-independent embeddings we use standard neural word2vec embeddings (Mikolov, Sutskever, et al., 2013) as implemented in the fastText library[8]. Note that the fastText embeddings use subword input that is suitable for morphologically rich languages (Bojanowski et al., 2017). As context-dependent embeddings we use ELMo (Peters et al., 2018)[9].

The dictionaries used in supervised methods are bilingual dictionaries extracted from wiktionary, using wikt2dict[10] tool (Acs, 2014). Some of the dictionaries, notably Croatian-Slovenian, were created with triangulation via English and are of low quality. We use two types of text embeddings evaluation scenarios, intrinsic and extrinsic. The intrinsic evaluation uses synthetic tasks and the evaluation metrics deal only with a given cross-lingual transformation. This type of evaluation is typically faster and can often be used to guide the construction of mappings. For non-contextual intrinsic evaluation of cross-lingual mapping, we use the dictionary induction and word analogy scenarios. The extrinsic evaluation scenarios use embeddings as inputs to downstream text mining task, in our case this is the named entity recognition scenario.

We evaluated cross-lingual mappings of non-contextual fastText embeddings and the contextual ELMo models we trained. We used supervised and unsupervised methods provided by MUSE and vecmap libraries. Four language pairs were considered for mapping, two between similar/related languages: Croatian-Slovenian (hr-sl) and Estonian-Finnish (et-fi), and two between more distant languages: English-Slovenian (en-sl) and English-Estonian (en-et). In each pair, we mapped embeddings using the first listed language as the source language and the second as the target language.

[8] `https://fasttext.cc/`

[9] Note that we did not include BERT, which is a multilingual model that does not need explicit alignment.

[10] `https://github.com/juditacs/wikt2dict`

4.1 Dictionary induction task

One of the standard benchmarks for intrinsic evaluation of cross-lingual embeddings is the dictionary induction task, also called a translation task. The goal is to find the correct translation for each given word. For example, given a pair of words from the Slovenian-English dictionary “drevo - tree”, we map the Slovene word “drevo” from Slovene to English and check for the vector among all English word vectors that is the closest to the mapped Slovenian vector for “drevo”. If the closest vector is “tree”, we count this as a success, else we do not. This measure is called precision@1 score or 1NN score: the number of successes, divided by the number of all examples, in this case dictionary pairs (see its formal definition below).

4.1.1 Experimental setting

For the dictionary induction task we used bilingual dictionaries extracted from wiktionary, using wikt2dict[11] tool (Acs, 2014). We split the dictionaries to training and evaluation set. We used the training set for the supervised cross-lingual mapping and the evaluation set for the dictionary induction task.

The evaluation of cross-lingual word embeddings shall measure the appropriateness of matching word pairs between two languages. The comparison metric shall give higher score to cross-lingual mappings where the nearest neighbor of a source word, in the target language, is more likely to have as the nearest neighbor this particular source word. For example, let us assume that we have a collection of word pairs from a dictionary and we want to use them to evaluate cross-lingual word embedding. We take a pair of words, s in a source language and t in a target language, and compute the cross-lingual mapping of the source word vector to the target embedding space. We search for the nearest words to that point. For the **iNN measure** (e.g., 1NN, 5NN, or 10NN), we calculate the percentage of correct target words found in the i-size neighbourhood of the mapped point.

This measure may be problematic, as nearest neighbors are by nature asymmetric: point y being a k-NN of point x does not imply that x is a k-NN of y. For example, some vectors, called hubs, are with high probability nearest neighbors of many other points, while others (anti-hubs) are not nearest neighbors of any point. To solve the problem of k-NN asymmetry, Conneau et al. (2018) proposed a metric, called **CSLS** (Cross-domain Similarity Local Scaling). The idea is to construct a bi-partite neighborhood graph, in which each word of a given dictionary is connected to its k nearest neighbors in the other language. Let x_s be a word in the source language and W be a cross-lingual mapping matrix which transforms x_s into the target embedding space Wx_s. Let $N_T(Wx_s)$ be the neighborhood on this bi-partite graph, associated with the mapped source word embedding Wx_s (i.e. in the target space). Note that all k elements of $N_T(Wx_s)$ are words from the target language. Similarly, let y_t be the word in the target language and $N_S(y_t)$ be the neighborhood associated with a word y_t of the target language. The mean similarity of a source embedding x_s to its target neighborhood is denoted as

$$r_T(Wx_s) = \frac{1}{k} \sum_{y_t \in N_T(Wx_s)} d_{cos}(Wx_s, y_t).$$

Similarly, we denote by $r_S(y_t)$ the mean similarity of the target word y_t to its neighborhood. These scores are computed for all source and target word vectors using an efficient nearest neighbors implementation, e.g., (Johnson et al., 2019). CSLS measure combines them into a similarity measure between mapped source words and target words

$$CSLS(Wx_s, y_t) = 2d_{cos}(Wx_s, y_t) - r_T(Wx_s) - r_S(y_t)$$

CSLS increases the similarity associated with isolated word vectors compared to iNN measure and decreases the similarity of vectors lying in dense areas.

[11] https://github.com/juditacs/wikt2dict

4.1.2 Results

In a translation or dictionary induction task, we check how many words are correctly translated by finding the closest word vector in another language, using precision@1 measure, where we used pre-computed fastText embeddings, trained on common-crawl corpora. Embeddings were in text format, where each word is given a vector, i.e. we know the vectors in advance, they are not computed on the spot.

The results are shown in Table 1. The results are similar for both retrieval metrics, CSLS and NN (1NN). In both cases, the vecmap mapping approach outperforms the MUSE approach in this task. The difference between supervised and unsupervised methods strongly depends on the similarity of languages in the mapped pair and on the quality of dictionaries. Unsurprisingly, unsupervised methods perform strongly on similar languages, and poorly on distant languages, especially on the English-Estonian language pair. In the Croatian-Slovenian pair, the poor quality of the dictionary results in poor results for the supervised method. In MUSE approach, the supervised method is only slightly better than the unsupervised method, while in vecmap approach the supervised method is even worse than the unsupervised method. In the Estonian-Finnish language pair, the supervised and unsupervised methods perform similarly; the MUSE supervised method performs better, while vecmap is better in an unsupervised setting.

Table 1: Comparison of cross-lingual mapping methods performed on translation/dictionary induction task for non-contextual fastText embeddings using iNN and CSLS metrics for translation retrieval. The result (precision@1 in %) of the best method for each language pair and for each retrieval metric is in bold.

Retrieval metric	Mapping method	en-sl	hr-sl	en-et	et-fi
NN	vecmap unsup	19.65	**34.21**	17.45	**48.54**
NN	vecmap sup	**30.21**	22.37	**35.85**	46.60
NN	MUSE unsup	13.54	28.72	9.14	40.59
NN	MUSE sup	23.49	29.43	28.32	47.52
CSLS	vecmap unsup	23.61	**36.51**	20.13	**53.10**
CSLS	vecmap sup	**33.14**	26.32	**36.64**	51.46
CSLS	MUSE unsup	16.15	29.79	11.11	46.53
CSLS	MUSE sup	24.47	31.91	31.36	52.48

Concerning other existing evaluations of cross-lingual embeddings on this task, Doval et al. (2019) tested supervised and unsupervised approaches of both vecmap and MUSE on a dictionary induction task. They mapped English embeddings to several different languages (Spanish, Italian, German, Farsi, Finnish, and Russian). They used embeddings trained on three different sources: Wikipedia, web corpora, and Twitter. In our work, we used fastText embeddings calculated on common crawl corpora, so below we summarize Doval et al. (2019) findings only for web corpora. In supervised approaches, vecmap and MUSE perform similarly, each is better on about half of the language pairs, with no clear divide. In unsupervised approaches, vecmap is slightly better. Doval et al. (2019) report negligible difference between unsupervised and supervised methods, except on Finnish and Russian as target languages, where supervised methods perform better. This contrasts with our findings, where unsupervised methods perform well only on similar languages, but not on distant ones. We also found vecmap consistently outperforming MUSE, but further testing is needed to confirm this finding.

Søgaard et al. (2018) tested MUSE unsupervised and supervised methods on English, Estonian, Finnish and a few other languages. For the supervised method, they used a small dictionary, which included only words that are identical in both languages. For Estonian-Finnish mapping, the unsupervised method outperformed the supervised method. In all other language pairs, the supervised method outperformed the supervised one. Notably, Søgaard et al. (2018) found that the unsupervised method performs poorly (scoring nearly 0.0) on English-Finnish and English-Estonian pairs. This is in slight contrast with our results, where unsupervised MUSE method performed much worse than other methods on English-Estonian (and also English-Slovenian) pairs, but much better than 0.0 score. Since they tested the results using various embeddings and confirmed the result in each case, we will further test our alignments with various dictionaries, since this is the only major difference between our approaches. We will report on these findings in the forthcoming deliverable D1.6 at M24.

4.2 Word analogy task

We evaluated cross-lingual mappings using an intrinsic evaluation approach based on our novel multilingual culture-independent analogy datasets (Ulčar & Robnik-Šikonja, 2019b). In the cross-lingual setting between two languages L_1 and L_2, the word analogy task (x is to y as a is to b) using a cross-lingual dataset is composed by matching each relation in one language with each relation from same category in the other language. An example of the composed cross-lingual dataset is shown in Table 2.

Table 2: An excerpt from the "city with river" category, showing two relations in English, two relations in Slovene and four relation pairs formed from them in a cross-lingual English-Slovene analogy dataset.

Relations: English			
Vienna	Danube		
Budapest	Danube		

Relations: Slovene			
Budimpešta	Donava		
Kairo	Nil		

Formed pairs (English-Slovene)			
Vienna	Danube	Budimpešta	Donava
Vienna	Danube	Kairo	Nil
Budapest	Danube	Budimpešta	Donava
Budapest	Danube	Kairo	Nil

Unfortunately, there are no existing intrinsic evaluation tasks for cross-lingual alignment of contextual embeddings; for cross-lingual contextual mappings the word analogy task is not adequate as it only contains words, without their context. We did, however, attempt to apply this task to a small number of words in context. We tested word analogies with contextual ELMo embeddings. The test set for contextual embeddings contained 65,000 words in a context. The embeddings were aligned using our locally isomorphic version of vecmap which gives multiple embeddings (one for each Voronoi cell). However, the coverage for the analogy task was low, only 10 analogies were tested and 2 of them matched. The preparation of a larger dictionary that will include all words from the analogy tasks is in progress.

Moreover, for intrinsic evaluation of cross-lingual contextual embeddings, we proposed a new dataset prepared as part of the SemEval 2020 challenge, addressing a new task named Graded Word Similarity in Context (GWSC); see *D1.3 Initial context-dependent and dynamic embeddings technology* for the description.

4.2.1 Experimental setting

In summary, the word analogy task is explored in two settings:

Monolingual setting. The goal of the word analogy task in a monolingual scenario is to find a term y for a given term x so that the relationship between x and y best resembles the given relationship $a : b$. For example, let the word pair $a : b$ be "Finland : Helsinki". The task is to find the term y corresponding to the relationship "Sweden : y", with the expected answer being $y =$ Stockholm. Monolingually, we measure the performance of embeddings in the target language L_2 after the cross-lingual mapping from L_1 (using all x, y, a, and b from the same language).

Cross-lingual setting. When evaluating the embeddings in a cross-lingual scenario, one pair of related words (a and b) is in one language and the other pair (x and y) from the same evaluation category (e.g., family relations, counties and capitals) is in another language. For example, given the pair a : b in English being “father : mother”, the task is to find the term y corresponding to the relationship “brat (brother) : y” in Slovene. The expected answer being $y =$ sestra (sister). Cross-lingually, we measure the performance of embeddings, where the first two words in an analogy pair are from the language L_1 and we search for the equivalent relation between the words from the analogy pair in language L_2.

We again use precision@1 score, meaning that the expected answer must be the closest of all possible words for the success. For both cases, we report the results on two language pairs, English-Slovenian (en-sl) and Croatian-Slovenian (hr-sl).

4.2.2 Results

We report our findings on the word analogy task below.

For each language pair, we tested four different mapping methods (vecmap and MUSE in both supervised and unsupervised mode), and evaluated the mapping on four analogy datasets. On monolingual analogy datasets (sl, en, hr) we evaluate the performance of the embeddings after applying the mapping (i.e. we measure how much the mapping hurts the monolingual performance in L_1). On cross-lingual analogy datasets (en-sl, sl-en, hr-sl, sl-hr) we evaluate how good relations in language L_1 match the equivalent relations in language L_2.

All methods perform comparably on monolingual datasets, without many differences between them (Table 3). On cross-lingual datasets the vecmap approach outperforms MUSE on this task, except for English monolingual experiment in English-Slovenian mapping, where MUSE has a slight edge. As in the dictionary induction task, vecmap unsupervised method outperforms vecmap supervised task in Croatian-Slovenian pair. In English-Slovenian pair the supvervised method gives better results.

Table 3: Comparison of cross-lingual mapping methods on the word analogy task, using non-contextual fastText embeddings. The best result (precision@1 in %) for each language pair is in bold. For example, for en-sl mapping and en dataset, we observed performance of English embeddings on English analogy dataset after mapping the vectors to a common space with Slovenian vectors.

Mapped pair	Analogy dataset	vecmap unsup	vecmap sup	MUSE unsup	MUSE sup
en-sl	sl	**43.42**	42.63	37.09	37.09
en-sl	en	63.82	67.29	**68.55**	**68.55**
en-sl	en-sl	29.98	**34.88**	19.17	24.79
en-sl	sl-en	58.59	**62.35**	45.14	53.28
hr-sl	sl	**46.50**	40.03	37.09	37.09
hr-sl	hr	**49.99**	43.42	41.70	41.70
hr-sl	hr-sl	**42.58**	32.92	30.63	32.22
hr-sl	sl-hr	**42.65**	33.25	30.63	30.92

In related experiments, Brychcín et al. (2019) tested cross-lingual mapping methods on the cross-lingual word analogy task in English, German, Italian, Spanish, Czech and Croatian. Authors were using different analogy dataset with different categories, so the results are not directly comparable. However, in all language pairs, where one of the languages was English, Brychcín et al. (2019) achieved much better results on a dataset of type xx-en than en-xx. The results on Czech-Croatian and Croatian-Czech datasets were similar, though. This agrees with our findings, where Slovene-Croatian and Croatian-Slovene perform very similarly, while Slovenian-English scores much better than English-Slovenian. Further, for many language pairs, they achieved better results on the cross-lingual dataset than on the monolingual dataset. In our tests, however, the results on monolingual datasets were always better than the results on cross-lingual datasets.

4.3 Named entity recognition

For extrinsic evaluation of cross-lingual mappings (both non-contextual and contextual), we use an adapted version of named entity recognition (NER) task, which is available for all EMBEDDIA languages. Note that the datasets used and the purpose of NER task in this report is different from the NER task in Deliverable *D2.2. Initial cross-lingual semantic enrichment technology*. In this report, we are only interested in NER for comparison of different alignment methods and not to maximally improve the NE recognition rate (e.g., we do not use any external information, fine-tuning of models etc).

NER is an information extraction task that seeks to locate and classify named entity mentions in unstructured text into pre-defined categories such as the person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc. The labels in the used NER datasets are simplified to a common label set of three labels (person name, location, organization), present in all the addressed working languages.

4.3.1 Experimental setting

We performed the NER task in three settings for each given language pair L_1-L_2.

1. For embeddings mapped from language L_1 to language L_2 (or both mapped into a common space), the first setting was training the NER model on L_1 train data and evaluating the model on L_2 test data.
2. The second setting was training the NER model on L_2 train data and evaluating the model on L_2 test data.
3. The third setting was training the NER model on L_1 train data and evaluating the model on L_1 test data.

For all three settings, we compared mapping methods with a reference value. The reference values allow for comparison with models without cross-lingual transfer, i.e. training and testing on instances from the same language. It was obtained by training the prediction model on L_2 train data and evaluating on L_2 test data for first and second setting, and by training on L_1 train data and evaluating on L_1 test data for the third setting. In all reference value cases no mapping was applied.

4.3.2 Results

For the NER task, we embedded the text using fastText or ELMo embeddings. For languages other than English we used pretrained high-quality ELMo models we created (Ulčar & Robnik-Šikonja, 2019a). For ELMo embeddings, we used the average of its three layers. We trained a neural network consisting of a single LSTM layer (using dropout), followed by a softmax layer. We trained the model for 5 epochs when using fastText embeddings, and 3 epochs when using ELMo embeddings. The input of the network were the embeddings, mapped using supervised and unsupervised methods provided by vecmap and MUSE. The embeddings were not fine-tuned to the task. In the case of ELMo, we only used vecmap, since it consistently outperformed MUSE methods on fastText embeddings (see Table 4)

- For unsupervised mapping, we followed the method of Schuster et al. (2019), briefly described in Section 3, except the mapping method itself was unsupervised, i.e. we did not use a dictionary.
- For supervised method we used our approach described in Section 3.1, using the same wiktionary dictionaries as in other tasks and OpenSubtitles parallel corpus[12] (Lison & Tiedemann, 2016) from

[12] https://www.opensubtitles.org/.

Table 4: Comparison of vecmap and MUSE supervised (sup) and unsupervised (unsup) methods for cross-lingual mapping on non-contextual fastText embeddings performed on the NER task. The best result for each training and evaluation language pair is in bold.

Setting	Mapped pair	Training language	Evaluation language	vecmap unsup	vecmap sup	MUSE unsup	MUSE sup
1	en-sl	en	sl	0.16	**0.27**	0.01	0.06
2	en-sl	sl	sl	**0.76**	0.75	0.69	0.69
3	en-sl	en	en	**0.30**	0.28	0.29	0.30
1	hr-sl	hr	sl	**0.58**	0.38	0.49	0.44
2	hr-sl	sl	sl	0.73	**0.77**	0.69	0.70
3	hr-sl	hr	hr	0.18	**0.19**	0.14	0.14
1	en-et	en	et	**0.21**	0.21	0.08	0.11
2	en-et	et	et	**0.28**	0.27	0.27	0.27
3	en-et	en	en	0.29	0.29	0.29	**0.29**
1	et-fi	et	fi	0.28	0.11	**0.37**	0.27
2	et-fi	fi	fi	0.69	**0.70**	0.69	0.69
3	et-fi	et	et	**0.27**	**0.27**	0.26	0.27

Opus web page[13].

- We applied the cross-lingual mapping of the ELMo embeddings on the fly, after producing the contextual embeddings of the dataset.

We present the results using the Macro F_1 score, that is an average of F_1 scores for each class we are trying to predict, excluding the class O (other, i.e. not a named entity). The results of fastText embeddings are shown in Table 4 and the results of ELMo embeddings in Table 5.

Table 5: Comparison of vecmap supervised (sup) and unsupervised (unsup) methods for cross-lingual mapping on contextual ELMo embeddings performed on the NER task. The best result for each language pair is in bold. The reference values (REF) represent the results on the evaluation data labeled on that line, but with a model trained on the same language as evaluation data and with no mapping applied, as explained in Section 4.3.

Setting	Mapped pair	Train language	Eval language	vecmap unsup	vecmap sup	REF
1	en-sl	en	sl	0.01	**0.59**	0.67
2	en-sl	sl	sl	0.73	**0.74**	0.67
3	en-sl	en	en	0.45	**0.46**	0.43
1	hr-sl	hr	sl	0.03	**0.66**	0.67
2	hr-sl	sl	sl	0.75	**0.77**	0.67
3	hr-sl	hr	hr	**0.54**	0.51	0.53

The results of fastText embeddings confirm what we have already observed in previous tasks: the vecmap methods, either supervised or unsupervised, outperform the MUSE methods. The results on ELMo embeddings, however, are more diverse. Models trained on one language and evaluated on another language perform very poorly when mapped with the unsupervised method and strongly when mapped with the supervised method. That is especially true on Croatian-Slovenian pair, where training on Croatian database, which is much smaller than Slovenian database, produces nearly identical results to training on Slovenian database when both are evaluated on the Slovenian evaluation data. In a monolingual setting (training and evaluating on the same language), all languages when mapped with either method perform comparably to the unmapped embeddings or even slightly better.

[13] http://opus.nlpl.eu

5 Graph-based language comparison

This section describes an approach, not anticipated in the EMBEDDIA proposal, but related to cross-lingual alignments. While this approach is not based on embeddings, it is relevant to the EMBEDDIA project as it may help to detect similar languages where the unsupervised cross-lingual embeddings could perform especially well. Modeling relations between languages can namely offer understanding of language characteristics and uncover similarities and differences between languages. This can help in understanding of language development over time and in improving cross-lingual natural language processing techniques.

A detailed description of the proposed approach to language similarity detection is contained in the appended paper published in Proceedings of the Statistical Language and Speech Processing (SLSP) 2019 conference (Škrlj & Pollak, 2019). In the paper we propose a novel approach to representing textual data as a directed, weighted network by the proposed text2net algorithm. Once presented as a network, computation of fast, network-topological metrics (such as metrics for network community structure) can be used for cross-lingual comparisons. We build networks from texts of nine selected languages contained in a large parallel corpus. On the constructed networks, we apply eight network topology metrics and use the results to interpret the relations between the languages. Community-based metrics, such as clustering coefficient, capture well-known differences between the languages, while others can be seen as novel opportunities for linguistic studies. The proposed method works on large corpora consisting of hundreds of thousands of aligned sentences even on an off-the-shelf laptop computer.

6 Associated outputs

The work described in this deliverable has resulted in the following resources:

Description	URL	Availability
ELMo embeddings	`hdl.handle.net/11356/1277`	Public (GPL v3)
Word analogy dataset	`hdl.handle.net/11356/1261`	Public (CC-BY-SA)
Crosslingual NER	`github.com/EMBEDDIA/crosslingual-NER`	To become public*
Vecmap changes	`github.com/EMBEDDIA/vecmap-changes`	To become public*

* Resources marked here as “To become public” are available only within the consortium while under development and/or associated with work yet to be published. They will be released publicly when the associated work is completed and published.

Parts of this work are also described in detail in the following publications, which are attached to this deliverable as appendices:

Citation	Status	Appendix
Ulčar, M., Robnik-Šikonja, M. (2019b). Multilingual culture-independent word analogy datasets. arXiv preprint arXiv:1911.10038.	Submitted (LREC 2020), available online	Appendix A
Škrlj, B., Pollak, S. (2019). Language comparison via network topology. In Proceedings of the international conference on statistical language and speech processing SLSP 2019 (pp. 112–123). Springer.	Published	Appendix B

7 Conclusions and further work

We have analyzed the existing cross-lingual embeddings approaches that map between monolingual embeddings. The results on all three analyzed tasks, dictionary induction, word analogy, and named entity recognition, show that the vecmap approach (supervised or unsupervised) is superior to the MUSE approach for both non-contextual and contextual embeddings. These findings are in agreement with recent results of Vulić et al. (2019), who also find the unsupervised mapping approaches inferior to at least weakly supervised approaches and attribute the reason to low quality of anchor points.

In further work, we will expand cross-lingual mappings in two directions: obtaining better anchor points for contextual embeddings and improving methods which drop the assumption of isomorphic monolingual embeddings in different languages. For the first direction, we are preparing much larger datasets of word-aligned sentences, which will provide high-quality anchor points for alignment of contextual embeddings. To avoid the assumption of isomorphic monolingual embeddings for different languages, we are going to continue our research on locally isomorphic transformations. These directions are in line with the recent findings that the assumption about isomorphism of embeddings in different languages leads to large errors for weakly related languages (Ormazabal et al., 2019). Dropping this assumption may lead to reasonable improvements as shown by the non-isomorphic method of Zhang et al. (2019). Further, we will expand the set of experiments to more languages, larger dictionaries, and more combinations of mapping methods, to more reliably contrast our findings with the results of Doval et al. (2019), Søgaard et al. (2018), and Brychcín et al. (2019).

However, we also have to take into account recent ever larger, ever more computationally demanding, and ever more successful language models, multilingual language models, and cross-lingual models (BERT, multilingual BERT and T5 by Google, UniLM by Microsoft, GPT-2 by OpenAI, Megatron by Nvidia, XLM and XLM-R by Facebook). Their recent results in text understanding and cross-lingual transfer show that it is essential to train huge neural network models on enormously large text collections and train them on many languages with absurd number of GPUs for a very long time. For example, at the time of this writing, a brand new state-of-the-art multilingual language model XLM-R (Conneau et al., 2019) uses 550 million neural network parameters, 2TB of training text from 100 languages, and was trained using 500 32GB-Nvidia-V100 GPUs for months. A single development cycle of any of the above mentioned models costs hundreds of thousands of Euros and is out of reach for most academic researchers. Nevertheless, as some of these models are publicly available, and can be fine-tuned and improved for specific languages and tasks, we will further investigate this path as one of the research directions (currently, we are working with multilingual BERT in the context of Tasks 1.2 and 1.3).

References

Acs, J. (2014). Pivot-based multilingual dictionary building using wiktionary. In *Proceedings of the ninth international conference on language resources and evaluation LREC.*

Aldarmaki, H., & Diab, M. (2019). Context-aware crosslingual mapping. In *Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies.*

Artetxe, M., Labaka, G., & Agirre, E. (2018a). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Thirty-second AAAI conference on artificial intelligence.*

Artetxe, M., Labaka, G., & Agirre, E. (2018b). 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)* (pp. 789–798).

Artetxe, M., & Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, *7*, 597–610.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Brychcín, T., Taylor, S., & Svoboda, L. (2019). Cross-lingual word analogies using linear transformations between semantic spaces. *Expert Systems with Applications*, *135*, 287-295.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., & Jégou, H. (2018). Word translation without parallel data. In *Proceedings of international conference on learning representation ICLR.*

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. 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)* (pp. 4171–4186).

Doval, Y., Camacho-Collados, J., Espinosa-Anke, L., & Schockaert, S. (2019). On the robustness of unsupervised and semi-supervised cross-lingual word embedding learning. *arXiv preprint arXiv:1908.07742*.

Dyer, C., Chahuneau, V., & Smith, N. A. (2013). A simple, fast, and effective reparameterization of IBM Model 2. In *Proceedings of NAACL-HLT* (pp. 644–648).

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259–284.

Lison, P., & Tiedemann, J. (2016). OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the 10th international conference on language resources and evaluation LREC.*

Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *arXiv preprint 1309.4168*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Ormazabal, A., Artetxe, M., Labaka, G., Soroa, A., & Agirre, E. (2019). Analyzing the limitations of cross-lingual word embedding mappings. In *Proceedings of the 57th annual meeting of the association for computational linguistics ACL* (pp. 4990–4995).

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing EMNLP* (pp. 1532–1543).

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 conference of the North American chapter of the Association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 2227–2237).

Schuster, T., Ram, O., Barzilay, R., & Globerson, A. (2019). Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. *arXiv preprint arXiv:1902.09492*.

Škrlj, B., & Pollak, S. (2019). Language comparison via network topology. In *Proceedings of the international conference on statistical language and speech processing SLSP 2019* (pp. 112–123). Springer.

Søgaard, A., Ruder, S., & Vulić, I. (2018). On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 778–788).

Søgaard, A., Vulić, I., Ruder, S., & Faruqui, M. (2019). *Cross-lingual word embeddings* (Vol. 12) (No. 2). Morgan & Claypool Publishers.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the eighth international conference on language resources and evaluation LREC'12.*

Ulčar, M., & Robnik-Šikonja, M. (2019a). High quality ELMo embeddings for seven less-resourced languages. *arXiv preprint arXiv:1911.10049*.

Ulčar, M., & Robnik-Šikonja, M. (2019b). Multilingual culture-independent word analogy datasets. *arXiv preprint arXiv:1911.10038*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vulić, I., Glavaš, G., Reichart, R., & Korhonen, A. (2019). Do we really need fully unsupervised cross-lingual embeddings? In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing EMNLP-IJCNLP* (pp. 4398–4409).

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP* (pp. 353–355).

Zhang, M., Xu, K., Kawarabayashi, K.-i., Jegelka, S., & Boyd-Graber, J. (2019). Are girls neko or shōjo? Cross-lingual alignment of non-isomorphic embeddings with iterative normalization. In *Proceedings of the 57th annual meeting of the association for computational linguistics ACL* (pp. 3180–3189).

Multilingual Culture-Independent Word Analogy Datasets

Matej Ulčar, Marko Robnik-Šikonja
University of Ljubljana, Faculty of Computer and Information Science,
Večna pot 113, SI-1000 Ljubljana, Slovenia
{matej.ulcar, marko.robnik}@fri.uni-lj.si

Abstract

In text processing, deep neural networks mostly use word embeddings as an input. Embeddings have to ensure that relations between words are reflected through distances in a high-dimensional numeric space. To compare the quality of different text embeddings, typically, we use benchmark datasets. We present a collection of such datasets for the word analogy task in nine languages: Croatian, English, Estonian, Finnish, Latvian, Lithuanian, Russian, Slovenian, and Swedish. We redesigned the original monolingual analogy task to be culturally independent and also constructed cross-lingual analogy datasets for the involved languages. We present basic statistics of the created datasets and their initial evaluation using fastText embeddings.

Keywords: word embeddings, analogy task, evaluation, less-resourced languages

1. Introduction

As an input, neural networks require numerical data. *Text embeddings* provide such an input, ensuring that relations between words are reflected in the distances in high-dimensional numeric space. There are many distinct models producing embedding vectors, using different specialized learning tasks, e.g., word2vec (Mikolov et al., 2013b), GloVe (Pennington et al., 2014), and FastText (Bojanowski et al., 2017). For training, the embeddings algorithms use large monolingual corpora that encode important information about word meaning as distances between vectors. In order to enable downstream machine learning on text understanding tasks, the embeddings shall preserve semantic relations between words, and this is true even across languages.

To compare the quality of different text embeddings, typically we use benchmark datasets. In this work, we present a collection of such datasets for the word analogy task in nine languages: Croatian, English, Estonian, Finnish, Latvian, Lithuanian, Russian, Slovenian, and Swedish. To make the datasets sensible for all languages, we designed the analogy task to be culturally independent.

The word analogy task was popularized by Mikolov et al. (2013c). The goal is to find a term y for a given term x so that the relationship between x and y best resembles the given relationship $a : b$. There are two main groups of categories: semantic and syntactic. To illustrate a semantic relationship, consider for example that the word pair $a : b$ is given as “Finland : Helsinki”. The task is to find the term y corresponding to the relationship “Sweden : y”, with the expected answer being $y =$ Stockholm. In syntactic categories, the two words in a pair have a common stem (in some cases even the same lemma), with all the pairs in a given category having the same morphological relationship. For example, given the word pair “long : longer”, we see that we have an adjective in its base form and the same adjective in a comparative form. The task is then to find the term y corresponding to the relationship “dark : y”, with the expected answer being $y =$ darker, i.e. a comparative form of the adjective dark.

In the vector space, the analogy task is transformed into vector arithmetic and we search for nearest neighbours, i.e. we compute the distance between vectors: d(vec(Finland), vec(Helsinki)) and search for word y which would give the closest result in distance d(vec(Sweden), vec(y)). In our dataset, the analogies are already pre-specified, so we are measuring how close are the given pairs.

The paper is split into further four sections. In Section 2., we describe the analogy task, its origin, culture-independent design, structure, and how it can be used as a benchmark for evaluation of embeddings in monolingual and cross-lingual setting. In Section 3., we present the creation of the actual monolingual and cross-lingual datasets and the process of their adaptation to all involved languages. We present statistics and initial evaluations of the produced datasets in Section 4.. Conclusion and plans for further work are described in Section 5..

2. Analogy task for embedding evaluations

We composed the analogy tasks involving nine languages: Slovenian, Croatian, Estonian, Finnish, Latvian, Lithuanian, Russian, Swedish, and English. The work is based on the English dataset by Mikolov et al. (2013a)[1]. Due to English- and US-centered bias of this dataset, we removed some categories and added or changed some of the others as described below. Our dataset was first written in Slovene language and then translated to other languages as explained in Section 3.3.. Following Mikolov et al. (2013a), we limit the analogies to single word terms, for example “New Zealand” is not a valid term for a country, since it consists of two words. Note that due to language differences, the produced datasets are not aligned across languages.

To assure consistency and allow the use of the datasets in cross-lingual analogies (described in section 2.1.), our datasets (even the English one) are somewhat different from the one by (Mikolov et al., 2013a). We removed or edited some categories and added new ones to avoid or limit English-centric bias in the following way.

[1] `http://download.tensorflow.org/data/questions-words`

- We merged two categories dealing with countries and their capitals (”common capital cities” and ”all capital cities”) into one category.

- We changed ”city in US state” category to ”city in country” and used mostly European countries with a better chance to appear in the corpora of respective languages.

- We removed the category ”currency”, as only a handful of currencies are present in news and text corpora with sufficient frequency.

- We added two new semantic categories, ”animals” and ”city with river” described below.

- We added a syntactic category comparing noun case relationships.

The resulting analogy tasks are composed of 15 categories: 5 semantic and 10 syntactic/morphological. The categories contained in our datasets are the following:

capitals and countries, capital cities in relation to countries, e.g., Paris : France,

family, a male family member in relation to an equivalent female member, e.g., brother : sister,

city in country, a non-capital city in relation to the country of that city, e.g., Frankfurt : Germany,

animals, species/subspecies in relation to their genus/familia, following colloquial terminology and relations, not biological, e.g., salmon : fish,

city with river, a city in relation to the river flowing through it, e.g., London : Thames,

adjective to adverb, an adverb in relation to the adjective it is formed from, e.g., quiet : quietly,

opposite adjective, the morphologically derived opposite adjective in relation to the base form, e.g., just : unjust, or honest : dishonest,

comparative adjective, the comparative form of adjective in relation to the base form, e.g., long : longer,

superlative adjective, the superlative form of adjective in relation to the base form, e.g., long : longest,

verb to verbal noun, noun formed from verb in infinitive form, e.g., to sit : sitting; in Estonian and Finnish - da infinitive and first infinitive forms are used respectively; in Swedish present participle that functions as noun is used in place of verbal noun,

country to nationality of its inhabitants, e.g., Albania : Albanians,

singular to plural, singular form of a noun in relation to the plural form of the noun, e.g., computer : computers; indefinite singular and definite plural are used in Swedish,

genitive to dative, a genitive noun case in relation to the dative noun case in respective languages, e.g. in Slovene ceste : cesti, singular is used for all words, except ”human” (or equivalent in other languages), which appears in both singular and plural; in Finnish and Estonian, dative has been replaced with the allative case, the category is not applicable to Swedish and English,

present to past, 3rd person singular verb in present tense in relation to 3rd person singular verb in past tense, e.g., goes : went; in Slovene, Croatian and Russian the masculine gender past tense is used, in other languages the ”simple” past tense/preterite is used,

present to other tense, 3rd person singular verb in present tense in relation to the 3rd person singular verb in various tenses, e.g., goes : gone; the other tense in Slovene, Croatian and Russian is feminine gender past tense, in Finnish, Estonian and English it is present/past perfect participle, in Swedish it is supine, in Latvian and Lithuanian it is future tense.

2.1. Cross-lingual analogies

Cross-lingual word embeddings have two or more languages in the same semantic vector space. Cross-lingual word analogy task has been proposed by Brychcín et al. (2019) as an intrinsic evaluation of cross-lingual embeddings. Following their work, we compose cross-lingual analogy datasets, so that one pair of related words is in one language and the other pair from the same category is in another language. For example, given the relationship in English father : mother, the task is to find the term y corresponding to the relationship brat (brother) : y in Slovene. The expected answer being $y =$ sestra (sister). We limited the cross-lingual analogies to the categories that all our languages have in common, i.e. excluding the last three named categories: genitive to dative, present to past and present to other tense.

3. Creation of datasets

Once the relations forming the analogies were prepared, we used them to form the actual monolingual and cross-lingual datasets. The process consisted of three steps. In Sections 3.1. and 3.2., we describe the creation of monolingual and cross-lingual datasets from the relations, and in Section 3.3., we explain the translation procedure which lead to creation of datasets in all involved languages.

3.1. Compiling monolingual dataset

The actual construction of the analogy dataset started by forming baseline relations for each category. First, we manually wrote the relations one per line, where each relation consists of two words. In the family category, an example of such a relation is “father, mother”. Next we combined all relations in each category with one another and wrote them in pairs, e.g., “father, mother, brother, sister” If a pair of relations share a common word, such a pair is excluded from the database. An example of forming relation pairs is shown in Table 1.

Table 1: An excerpt from the "city with river" category, showing four relations and five relation pairs formed from them. The first two listed relations do not form a pair with each other, because they share a common word (Danube).

Relations			
Vienna	Danube		
Budapest	Danube		
Cairo	Nile		
Paris	Seine		

Formed pairs			
Vienna	Danube	Cairo	Nile
Vienna	Danube	Paris	Seine
Budapest	Danube	Cairo	Nile
Budapest	Danube	Paris	Seine
Cairo	Nile	Paris	Seine

3.2. Cross-lingual datasets

Cross-lingual analogies described in Section 2.1. are compiled in a similar manner. Consider a language pair $L_1 - L_2$. From the same one-relation-per-line files shown in the upper part of Table 1, we combine all relations in a category in such a way that one relation from language L_1 and one relation from language L_2 form a pair. L_1 relations are on the left-hand side, and L_2 relations are on the right-hand side. An example of forming cross-lingual relation pairs is shown in Table 2 for English-Slovene language pair. The same rules for excluding pairs with common words apply, except that we do not consider translations of the same term as the same word, e.g., "Nile" (in English) and "Nil" (its Slovene equivalent) in the same entry are allowed, but using "Nile" twice is disallowed.

Table 2: An excerpt from the "city with river" category, showing two relations in English, two relations in Slovene and four relation pairs formed from them in a crosslingual English-Slovene analogy dataset.

Relations: English			
Vienna	Danube		
Budapest	Danube		

Relations: Slovene			
Budimpešta	Donava		
Kairo	Nil		

Formed pairs (English-Slovene)			
Vienna	Danube	Budimpešta	Donava
Vienna	Danube	Kairo	Nil
Budapest	Danube	Budimpešta	Donava
Budapest	Danube	Kairo	Nil

3.3. Translation procedure

When the first dataset in Slovene was formed, we translated it into other languages (including English). We used various tools to help us translate Slovenian dataset to the other languages. For the geographic data, i.e. names of countries, cities and rivers, we used the titles of equivalent Wikipedia articles or data from Wikipedia lists, such as the list of capital cities. If an entity had a name consisting of more than one word in another language, it was either skipped or replaced by another entity with subjectively similar location and/or importance. The same was done in cases where we would have a relation of type "x : x", which is nonsensical. For example, in Lithuanian Algeria and its capital Algiers are both called "Alžyras". So we replaced it with "Damaskas : Sirija" (in English this would correspond to "Damascus : Syria").

For non-geographic words, we mostly used Babelnet[2] and Wiktionary[3] to find the translations. In the latter, we mostly relied on conjugation and declination tables of our key words. Wiktionary was also used for finding new examples for relations in syntactical categories, to replace those for which a translation was either impossible or could not be found. This was most often the case in all the categories operating with adjectives. An example of an impossible translation is the Slovene relation "drag : dražji". Its English translation is "expensive : more expensive". Since we are limited to single-word terms, we discarded that translation and replaced such a relation with another one, with either a similar meaning "costly : costlier", or a completely different one, like "high : higher", provided it does not already appear in the dataset.

English and Swedish languages do not have noun cases or rather only have genitive case (in addition to nominative) in a very limited sense. We decided to exclude the "genitive to dative" category for these two languages. Further more, while Finnish and Estonian have many noun cases, none of those cases is dative. We exchanged dative in this category with allative case, which mostly covers the same role.

For two categories, "city with river" and in a smaller part "city in country" we intentionally varied the entries across languages more than in other categories, where it was only done so out of necessity. We felt certain relations are too locally specific to frequently (or at all) appear in other language corpora. We removed most of such relations in other languages and tried to replace them with other relations more geographically local to that language, in order to keep the number of different countries or rivers high. Majority of the relations in these two categories is still the same for all languages.

The translated relations were checked by native speakers of each language and corrected where deemed necessary.

4. Statistics and evaluation

In this section, we first present relevant statistics of the created datasets, followed by their evaluation using fastText embeddings.

[2] https://babelnet.org/

[3] https://wiktionary.org

4.1. Statistics

The original English analogy dataset by Mikolov et al. (2013a) contains 19,544 relations, but uses slightly different categories to our datasets. As explained above, we translated the Slovene dataset into all other languages to keep datasets similar across languages, especially for the use in cross-lingual analogy tasks. The number of obtained analogy pairs in monolingual datasets is between 18,000 to 20,000 per language. The exact numbers differ from language to language based on the validity of categories and availability of sensible examples in each category. The exact numbers for all languages are shown in Table 3.

Table 3: The sizes of the constructed monolingual word analogy datasets expressed as numbers of pairs for each language.

Language	Size
Croatian	19416
English	18530
Estonian	18372
Finnish	19462
Latvian	20138
Lithuanian	20022
Russian	19976
Slovene	19918
Swedish	18480

The number of pairs in cross-lingual datasets is smaller, because some categories were omitted. We created cross-lingual datasets for all 72 language pairs. The exact sizes of datasets for a few selected pairs are shown in Table 4.

Table 4: The sizes of a few constructed cross-lingual word analogy datasets expressed as numbers of pairs for each language.

Language pair	Size
Croatian-English	17667
Croatian-Slovene	17449
English-Slovene	17964
Estonian-Finnish	16809
Estonian-Slovene	17110
Finnish-Swedish	17600
Latvian-Lithuanian	18056

Not all categories are equally represented, some have much more relation pairs than others. We tried to downplay the importance of the category “capitals and countries”, which is very prominent in the dataset by Mikolov et al. (2013a), however, it is still by far the largest category in our dataset. Some categories are necessarily small, like “family”, since the number of terms for family members is relatively small. That is especially true for languages from northern Europe, so we also included plural terms and some non-family members in that category, like a relation “king : queen”. The number of analogy pairs per category (averaged over all languages) is shown in Table 5.

Table 5: Average size in number of pairs for each category in the monolingual word analogy datasets.

Category	Average size
Capitals and countries	5701
Family	482
City in country	2880
Animals	1440
City with river	701
Adjective to adverb	873
Opposite adjective	498
Comparative adjective	866
Superlative adjective	823
Verb to verbal noun	415
Country to nationality	924
Singular to plural	1519
Genitive to dative	1356
Present to past	607
Present to other tense	601

4.2. Evaluation

We evaluated the analogy datasets using the fastText (Bojanowski et al., 2017) embeddings[4]. The fastText embeddings use subword inputs which are suitable also for morphologically rich languages, we are working on. We limited the evaluation to the first 200,000 word vectors (i.e. 200,000 most frequent tokens) from the embeddings of each language. Not all analogy pairs can be evaluated in that way, since some words do not appear among the first 200,000 words. The amount of pairs that are covered (i.e. all four words from the analogy are among the most frequent 200,000 words) for each language is shown in the Table 6.

Table 6: Percentage of constructed analogy pairs covered by the first 200,000 word vectors from common crawl fastText embeddings.

Language	Coverage (%)
Croatian	81.67
English	97.05
Estonian	82.56
Finnish	63.97
Latvian	73.60
Lithuanian	77.66
Russian	62.53
Slovene	86.70
Swedish	82.44

We evaluated the relations that are completely contained in the first 200,000 fastText vectors. Given a pair of relations “a : b $\approx$ c : d”, we searched for the closest word vector to the vector $b - a + c$. We report the number of times the

[4] https://fasttext.cc/

closest word vector was vector of the word d. The results for all languages per category are shown in Table 7.

The results show that not all relations are recognized with the same accuracy across languages, the differences being large and surprising in some cases. This hints that there is a considerable space for improvement in construction of word embeddings.

5. Conclusion

We prepared word analogy datasets for nine languages: Croatian, English, Estonian, Finnish, Latvian, Lithuanian, Russian, Slovenian, and Swedish. The datasets are suitable for evaluation of monolingual embeddings as well as cross-lingual mappings. We describe the choice of 15 categories, 5 semantic and 10 syntactic, and an effort to make them language and culture independent. While the resulting datasets in nine languages are not aligned, they are nevertheless compatible enough to allow creation of cross-lingual analogy tasks for all 72 language pairs. We present basic statistics of the created datasets and their initial evaluation using fastText embeddings. The results indicate large differences across languages and categories, and show that there is a substantial room for improvement in creation of word embeddings that would better represent relations present in the language as distances in vector spaces.

As further challenge we see creation of similar intrinsic tasks for the evaluation of contextual embeddings.

The datasets of word analogy tasks for all nine languages and all language combinations will be deposited to Clarin repository[5] by the time of the final version of this paper.

Acknowledgements

We thank to native speakers that checked the respective monolingual analogies: Kristiina Vaik (Estonian), Dace Linde (Latvian), Milda Dailidėnaitė (Lithuanian), and Jessica Lindström (Finnish, Swedish)

The work was partially supported by the Slovenian Research Agency (ARRS) core research programme P6-0411. This paper is supported by European Union's Horizon 2020 research and innovation programme under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The results of this publication reflects only the authors' view and the EU Commission is not responsible for any use that may be made of the information it contains.

6. Bibliographical References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Brychcín, T., Taylor, S., and Svoboda, L. (2019). Cross-lingual word analogies using linear transformations between semantic spaces. *Expert Systems with Applications*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint 1301.3781*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint 1309.4168*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013c). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[5] Here comes a link to the repository.

Table 7: FastText evaluation scores in % of correctly predicted relation pairs, i.e. how often was the vector d the closest to the vector $b - a + c$, given a relation pair $a : b \approx c : d$.

Category	sl	en	hr	et	fi	lv	lt	sv	ru
capital	28.13	95.23	34.9	43.33	79.09	45.9	53.75	88.38	81.26
family	38.77	92.03	52.94	51.84	67.14	54.39	54.78	68.1	62.09
city in country	45.44	89.92	47.21	46.34	85.31	56.66	63.25	90.18	95.26
animals	1.13	11.72	0.85	0.6	22.5	1.93	1.43	10.88	18.44
city-river	5.92	44.81	3.8	9.19	14.81	2.96	8.33	30.91	14.36
adjective-adverb	37.93	31.69	36.15	51.98	64.83	53.22	60.58	84.33	29.31
opposite	55.3	57.89	62.73	52.38	52.22	72.73	75.26	21.9	0
comparative	33.62	96.88	40.6	72.36	84.77	73.93	57.23	78.82	41.88
superlative	26.28	97.31	24.89	31.37	75.69	18.75	62.45	51.08	51.92
verbal noun	70	82.37	64.74	93.27	98.57	76.05	69.85	44.58	57.14
nationality	34.92	56.56	54.1	54.5	81.58	46.75	53.85	82.67	93.51
singular-plural	33.54	94.2	36.77	74.29	92.14	42.38	52.55	44.76	57.35
gentitive-dative	30.02	N/A	33.69	73.55	64.67	45.06	37.75	N/A	41.8
present-past	56.32	76.5	66.02	90.5	86.36	79.17	71.56	97.23	81.05
present-other	59.09	32.55	66.08	69.83	86.58	72.38	74.29	90.94	82.63

Language comparison via network topology

Blaž Škrlj[1,2] and Senja Pollak[2,3]

[1] Jožef Stefan International Postgraduate School
[2] Jožef Stefan Institute, Slovenia
[3] Usher Institute, Medical School, University of Edinburgh, Edinburgh
{blaz.skrlj,senja.pollak@ijs.si}

The final authenticated publication is available online at 10.1007/978-3-030-31372-2_10.

Abstract. Modeling relations between languages can offer understanding of language characteristics and uncover similarities and differences between languages. Automated methods applied to large textual corpora can be seen as opportunities for novel statistical studies of language development over time, as well as for improving cross-lingual natural language processing techniques. In this work, we first propose how to represent textual data as a directed, weighted network by the text2net algorithm. We next explore how various fast, network-topological metrics, such as network community structure, can be used for cross-lingual comparisons. In our experiments, we employ eight different network topology metrics, and empirically showcase on a parallel corpus, how the methods can be used for modeling the relations between nine selected languages. We demonstrate that the proposed method scales to large corpora consisting of hundreds of thousands of aligned sentences on an of-the-shelf laptop. We observe that on the one hand properties such as communities, capture some of the known differences between the languages, while others can be seen as novel opportunities for linguistic studies.

Keywords: Computational typology · cross-linguistic variation · network theory · language modeling · comparative linguistics · graphs · language representation

1 Introduction and related work

Understanding cross-linguistic variation has for long been one of the foci of linguistics, addressed by researchers in comparative linguistics, linguistic typology and others, who are motivated by comparison of languages for genetic or typological classification, as well as many other theoretical or applied tasks. Comparative linguistics seeks to identify and elucidate genetic relationships between languages and hence to identify language families [26]. From a different angle, linguistic typology compares languages to learn how different languages are, to see how far these differences may go, and to find out what generalizations can be made regarding cross-linguistic variation on different levels of language structure and aims at mapping the languages into types [6]. The availability of large electronic text collections, and especially large parallel corpora, have offered new possibilities for computational methodologies that are developed to capture cross-linguistic variation. This work falls under computational typology [13,1], an emerging field with the goal of understanding of the differences between languages via computational (quantitative) measures. Recent studies already offer novel insights into the inner structure of languages with respect to various sequence fingerprint comparison metrics, such as for example the Jaccard measure, the intra edit distance and many other boolean distances [21]. Such comparisons represent e.g., sentences as vectors, and evaluate their similarity using plethora of possible metrics. Albeit useful, vector-based representation of words, sentences or broader context does not necessarily capture the context relevant to the task at hand and the overall structure of a text collection. Word or sentence embeddings, which recently serve as the language representation workhorse, are not trivial to compare across languages, and can be expensive to train for new languages and language pairs (e.g., BERT [8]). Further, such embeddings can be very general, possibly problematic for use on smaller data sets and are dependent on input sequence length.

In recent years, several novel approaches to computational typography have been applied. For example, Bjerva et al. [2] compared different languages based on distance metrics computed on universal dependency trees [19]. They discuss whether such language representations can model geographical, structural or family distances between languages. Their work shows how a two layer LSTM neural network [12] represents the language in a structural manner, as the embeddings mostly correlate with structural properties of a language. Their main focus is thus on explaining the structural properties of neural network word embeddings. Algebraic topology was also successfully used to study syntax properties by Port et al. [20]. Similar efforts of statistical modelling of language distances were previously presented in e.g., [14] who used Kolmogorov complexity metrics.

In contrast, we propose a different approach to modeling language data. The work is inspired by ideas of node representation as seen in contemporary geometric and manifold learning [10] and the premises of computational network theory, which studies the properties of interconnected systems, found within virtually every field of science [27]. Various granularities of a given network can be explored using approaches for community detection, node ranking, anomaly identification and similar [9,15,5]. We demonstrate that especially information flow-based community detection [7] offers interesting results, as it directly simulates information transfer across a given corpus. In the proposed approach, we thus model a corpus (language) as a single network, exposing the obtained representation to powerful network-based approaches, which can be used for language comparison (as demonstrated in this work), but also for e.g., keyword extraction (cf. [4] who used TopicRank) and potentially also for representation learning and end-to-end classification tasks.

The purpose of this work is twofold. First, we explore how a text can be transformed into a network with minimal loss of information. We believe that this powerful and computationally efficient text representation that we name *text2net*, standing for text-to-network transformation, can be used for many new tasks. Next, we show how the obtained networks can be used for cross-lingual analysis across nine languages (36 language pairs).

This work is structured as follows. In Section 2 we introduce the networks and the proposed text2net algorithm. Next, we discuss network-topological metrics (Section 3) that we use for the language comparison experiment in Section 4. The results are presented in Section 5, followed by discussion and conclusions in Section 6.

2 Network-based text representation

First, we discuss the notion of networks, and next present our text2net approach.

2.1 Networks

We first formally define the type of networks considered in this work.

Definition 1 (Network). *A network is an object consisting of nodes, connected by arcs (directed) and /or edges (undirected). In this work we focus on directed networks, where we denote with $G = (N, A)$ a network G, consisting of a set of nodes N and a set of arcs $A \subseteq N \times N$ (ordered pairs).*

Such simple networks are not necessarily informative enough for complex, real world data. Hence, we exploit the notion of weighted directed networks.

Definition 2 (Directed weighted network). *A directed weighted network is defined as a directed network with additional, real-valued weights assigned to arcs.*

Note that assigning weights to arcs has two immediate consequences: arcs can easily be pruned (using a threshold), and further, algorithms, which exploit arc weights can be used. We continue to discuss how a given text is first transformed into a directed weighted network G.

2.2 text2net algorithm

Given a corpus T, we discuss the mapping text2net : $T \rightarrow G$. As text is sequential, the approach captures global word neighborhood, proceeding as follows:

1. Text is first tokenized and optionally stemming, lemmatization and other preprocessing techniques are applied to reduce the space of words.

2. text2net traverses each input sequence of tokens (e.g., words, or lemmas or stems depending on Step 1), and for each token (node) stores its successor as a new node connected with the outbound arc. This step can be understood as breaking the the text into triplets, where two consecutive words are connected via a directed arc (therefore preserving the sequential information).
3. During construction of such triplets, arcs commonly repeat, as words often appear in same order. Such repetitions are represented as arc weights. Weight assignment can depend on the arc type. For this purpose, we introduce a mapping $\rho(a) \rightarrow \mathbb{R}; a \in A$ (A is the set of arcs), a mapping which assigns a real value to a given arc with respect to that arc's properties.
4. Result is a weighted, directed network representing weighted token co-occurrence.

The algorithm can thus formally be stated as given in Algorithm 1. The key idea is to incrementally construct a network based on text, while traversing the corpus *only once* (after potential selected preprocessing steps).

We next discuss the text2net's computational complexity. To analyze it, we assume the following: the text corpus T is comprised of s sentences. In terms of space, the complexity can be divided into two main parts. First, the memory needed to store the sentence being currently processed and the memory for storing the network. As the sentences can be processed in small batches, we focus on the spatial complexity of the token network. Let the corpus consist of t tokens. In the worst case, all tokens are interconnected and the spatial complexity is quadratic $\mathcal{O}(t^2)$. Due to Zipf's law networks are notably smaller as each word is (mostly) connected only with a small subset of the whole vocabulary (heavy tailed node degree distribution). The approach is thus both spatially, as well as computationally efficient, and can easily scale to corpora comprised of hundreds of thousands of sentences.

In terms of hyperparameters, the following options are available (offering enough flexibility to model different aspects of a language, rendering text2net suitable as the initial step of multiple down-stream learning tasks):

- minimum sentence length considered for network construction (t_s),
- minimum token length (t_l),
- optional word transformation (e.g., lemmatisation) (f),
- optional stopwords or punctuation to be removed (σ),
- arc weight assignment function (ρ) (e.g., co-occurrence frequency),
- a threshold for arc prunning based on weights (θ).

3 Considered network topology metrics

In this section we discuss the selected metrics that we applied to directed weighted networks. The metrics vary in their degree of computational complexity.

Number of nodes. The number of nodes present in a given network.
Number of edges. The number of edges in a given network.

InfoMap communities. The InfoMap algorithm [22] is based on the idea of minimal description length of the walks performed by a random walker traversing the network. It obtains a network partition by minimizing the description lengths of random walks, thus uncovering dense regions of a network, which represent communities. Once converged, InfoMap yields the set of a given network's nodes N partitioned into a set of partitions which potentially represent functional modules of a given network.

Average node degree. How many in- and out connections a node has on average. For this metric, networks were considered as undirected. See below:

$$\text{AvgDeg} = \frac{1}{|N|}\sum_{n\in N} \text{deg}_{in}(n) + \text{deg}_{out}(n)$$

.

Network density. The network density represents the percentage of theoretically possible edges. This metric is defined as:

$$\text{Density} = \frac{|A|}{|N|(|N|-1)};$$

where $|A|$ is the number of arcs and $|N|$ is the number of nodes. This measure represents more coarse-grained clustering of a network.

Clustering coefficient. This coefficient is defined as the geometric average of the subnetwork edge weights:

$$\text{ClusCoef} = \frac{1}{|N|}\sum_{u\in N}\left(\frac{1}{deg(u)(deg(u)-1)}\sum_{vw}\sqrt[3]{(\hat{w}_{uv}\hat{w}_{uw}\hat{w}_{vw})}\right);$$

Algorithm 1: text2net algorithm.

```
Data: Text corpus $T$ (of documents $d_1 \dots d_n$), empty weighted network $G$
Parameters : Minimum number of tokens per sentence $t_s$, Minimum token
             length $t_l$, word transformation function $f$, stopwords $\sigma$, weight
             prunning threshold $\theta$, frequency weight function $\rho$
Result: A weighted network $G$
for $d \in T$ do
    orderedTokens := getTokens($d$, $t_l$,$t_s$,$f$,$\sigma$);        ▷ Get token sequence.
    for $q_i \in$ orderedTokens do
        arc := ($q_i$,$q_{i+1}$);                                  ▷ Construct an arc.
        addToNetwork($G$, arc);                                ▷ Construct the network.
        if arc $\in$ current set of arcs of $G$ then
            update arc's weight via $\rho$;                        ▷ Update weights.
        end
    end
end
G := prunenetwork($G$,$\theta$);                               ▷ Prune the network.
return $G$
```

here, $\hat{w}_{vw}$ for example represents the weight of the arc between nodes v and w. The $deg(u)$ corresponds to the u-th node's degree. Intuitively, this coefficient represents the number of closed node triplets w.r.t. number of all possible triplets. The higher the number, the more densely connected (clustered) the network. See [3] for detailed description of the metrics above.

4 Language comparison experiments

In this section we discuss the empirical evaluation setting, where we investigated how the proposed network-based text representation and network-topology metrics can be used for the task of language comparison. We use the parallel corpus (i.e., corpus of aligned sentences across different languages) from the DGT corpus, i.e. Directorate-General for Translation translation memory, provided by Joint Research Centre and available in OPUS [25]. We selected nine different languages: EN – English, ES – Spanish, ET – Estonian, FI – Finish, LV – Latvian, NL – Dutch, PR – Portugese, SI – Slovene, SK – Slovak, covering languages from different historical origins and language families: Romance languages (PT, ES), Balto-Slavic languages including Slavic (SI, SK) and Baltic (LV) language examples, Germanic langauges (EN, NL), as well as Finnic languages from Uralic family (FI, ET). The selected languages have also different typological characteristics. For example in terms of morphological typology, EN can be considered as mostly analytic, while majority of others are synthetic languages, where for example FI is considered as agglutinative, while Slavic languages are fusional as they are highly inflected.

The goal of the paper was to use the network topology metrics for langauge comparison. We considered all the pairs between the selected languages, resulting in 36 comparisons for each network-based metric. From the parallel corpus we sampled 100,000 sentences for each language, resulting in 900,000 sentences, which match across languages.

From each language, we constructed a network using text2net with following parameters: the minimum number of tokens per sentence (t_s) was set to 3, the minimum length of a given token (t_l) to 1, the word transformation function transformed words to lower-case, no lemmatisation was used, and punctuation was removed. We defined $\rho(\text{arc}) = 1$.

We compared the pairs of languages as follows. For each of the two languages, we transformed the text into a network. The discussed network topology metrics were computed for each of the two networks. Differences between the metrics' values are reported in tabular form (Table 1), as well as visualized as heatmaps (Figure 1). In the latter, the cells are colored according to the *absolute* difference in a given metric for readability purposes. Thus, the final result of the considered analysis are differences in a selected network topology metric. The selected results were further visualized in Figure 2.

We used NLTK [16] for preprocessing, Py3plex [23], NetworkX [11], Cytoscape [24] for network analysis and visualizatino and Pandas for numeric comparisions [17]. Full code is available at: `https://github.com/SkBlaz/language-comparisons`.

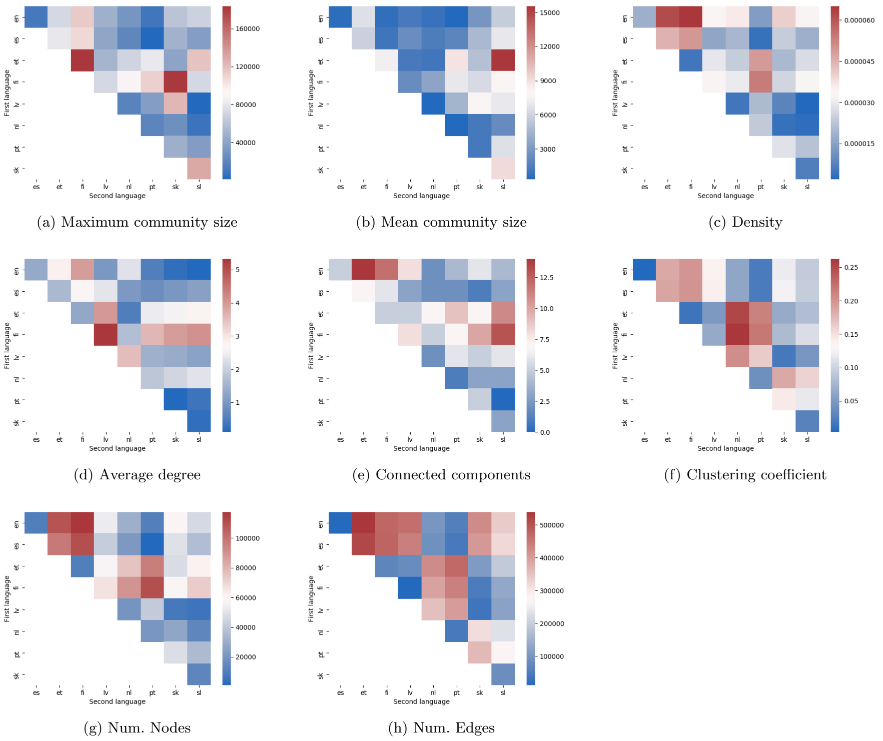
While we do not have full linguistic hypotheses about the expected mapping of the linguistic characteristics and the topological metrics, we believe that the network-based comparisons should show differences between the languages. For example, the number of nodes might capture linguistic properties, such as inflectional morphology, where we could expect that morphologically rich languages would have more nodes. Number of edges might capture linguistic properties, such as the flexibility of the word order. The other measures are less intuitive and will be further investigated in future work. However, we believe that more complex the language (including aspects of morphology richness and word order flexibility), the richer the corresponding network's structure, while the number of connected components might offer insights into general dispersity of a given language, and could pinpoint grammatical differences if studied in more detail. Also clustering coefficient might be dependent on how fixed is the word order of a given language. None of the above has been systematically investigated, and the hypothesis is, that differences between languages will have high variability and show already known, as well as novel groupings of the languages.

5 Results

In this section we present the results of cross-lingual comparison. The inter-language differences in tabular format are given in Table 1. The measures given in the table are the differences in: #Nodes — the number of nodes, #Edges — the number of edges, Mean degree — mean node degree, Density — network density as defined in Section 3, MaxCom — maximum community size, MeanCom — mean community size, both computed using InfoMap communities, Clustering — clustering coefficient and CC — the number of connected components. The differences in the table are presented in L2-L1 absolute differences, while for nodes and edges we also present the differences as relative percentages of the e.g., number of nodes of the second language w.r.t the number of nodes of the first language[4]. It can be observed that some language pairs differ substantially even if only node counts are considered, where EN-FI is the pair with the largest difference, which is not surprising. English is for example an analytical language, while Finnish agglutinative with very rich morphology. Further, some of the metrics indicate groupings, which can be further investigated using heatmaps and direct visualization of language-language links.

From heatmaps shown in Figure 1, where colors of individual cells represent differences between a given metric's values across languages, we can make several interesting observations. Based on Num. of nodes, FI and ET are very similar, and the most different to other languages. Both are agglutinative languages and part of the Uralic language family. In terms of Num. of edges, the largest differences are between ET and EN, while the most similar are LV and FI; in pairwise comparison with EN, we can see that PT, ES and NL have similar

[4] For nodes $N_{\text{diff}} = \frac{100 \cdot |N_2|}{|N_1|}$, and for edges $E_{\text{diff}} = \frac{100 \cdot |E_2|}{|E_1|}$; the first language's values are compared against the second language's values.



(a) Maximum community size

(b) Mean community size

(c) Density

(d) Average degree

(e) Connected components

(f) Clustering coefficient

(g) Num. Nodes

(h) Num. Edges

Fig. 1: Pairwise language comparison via various network-topological metrics. Cells represent the absolute differences between metrics of individual text-derived networks. Red regions represent very different networks,and blue very similar ones.

statistics, which are all languages from Germanic (NL) or Romanic family. We believe that some measures could also indicate groupings based on morphological or other typological properties beyond the currently known ones. For example, Max. community size on one hand points FI and ET as very different, as well as SI and SK (where in both pairs the two languages are belonging to the same language family), but on the other hand PT and ES are very similar. Further, Clustering coefficient yields insights into context structure and similar properties of groupings of basic semantic units, such as words, where high similarity between ES and PT, as well as SI and SK can be observed. Finally, the number of connected components offers insights into general dispersity of a given language, and could pinpoint grammatical differences if studied in more detail. Again, we see the most remarkable differences between EN and FI and ET, but also FI and SI, while Romanic and Germanic languages are more similar. There are many open questions. E.g., which linguistic phenomena make EN-FI being quite different in Average degree, while FI-NL are relatively similar (despite EN and NL being in the same language group)?

Clustering coefficient is also shown in an alternative visualisation, i.e. in a colored network in Figure 2. Here, we consider Clustering coefficient metric, where we adjust the color so that it represents only very similar languages (low absolute difference in the selected metric). We selected this metric, as the heatmap yielded the most block-alike structure, indicating strong connections between subsets of languages. We can see that Balto-Slavic and Finnic languages group together, while Germanic and Romanic form another group. Finally, we visualized the English corpus network in Figure 3. Colored parts of the network correspond to individual communities. It can be observed that especially the central part of the network contains some well defined structures (blue and red). The figure also demonstrates, why various network-topological metrics were considered, as from the structure alone, no clear insights can be obtained at such scale.

Table 1: Differences between selected network-topology metrics across languages. The values are computed as L2-L1, or reported as L2 relative to L1.

Language pair	#Nodes	#Edges	Mean degree	Density ($\cdot 10^{-4}$)	MaxCom	MeanCom	Clustering	CC	N_{diff} (%)	E_{diff} (%)
en-es	10232	15251	-1.34	-0.18	11100	538.53	-0.00	5	110.90	101.71
en-et	108986	539449	-2.85	-0.62	-72382	-6578.80	-0.19	14	233.53	187.45
en-fi	117623	474376	-3.96	-0.65	114803	649.71	-0.20	12	249.01	179.60
en-lv	53162	464366	1.02	-0.34	48411	-1208.95	-0.14	8	164.35	177.99
en-nl	30786	99189	-2.30	-0.36	30839	836.35	0.06	2	138.10	115.73
en-pt	10778	56039	-0.51	-0.14	10249	-366.93	0.02	4	113.07	107.48
en-sk	59715	425657	-0.22	-0.41	60709	2757.51	-0.12	6	174.20	172.24
en-sl	46764	337421	-0.11	-0.34	-68833	-5693.41	-0.09	4	156.30	154.82
es-et	97822	518349	-1.62	-0.35	-80506	-5877.34	-0.19	7	210.57	184.29
es-fi	110493	479001	-2.70	-0.49	108672	1004.91	-0.20	6	224.53	176.58
es-lv	42062	442253	2.34	-0.16	42066	2382.83	-0.14	3	148.20	174.99
es-nl	21501	88846	-1.04	-0.20	21235	1433.40	0.06	-2	124.52	113.78
es-pt	971	43922	0.84	0.04	1232	1922.25	0.02	-2	101.96	105.67
es-sk	49382	406578	0.99	-0.20	49740	4703.95	-0.12	1	157.08	169.34
es-sl	36317	321960	1.17	-0.18	36362	6935.34	-0.10	-3	140.94	152.21
et-fi	10262	-68268	-1.32	-0.05	183810	7318.28	-0.01	5	106.63	95.82
et-lv	-57119	-80883	4.01	0.29	-51457	1237.71	0.05	-5	70.38	94.95
et-nl	-75247	-424500	0.50	0.24	-69698	-1081.51	0.25	-7	59.14	61.74
et-pt	-96441	-471464	2.45	0.49	81260	8871.81	0.21	-9	48.42	57.34
et-sk	-47901	-109523	2.56	0.20	-40340	5107.84	0.07	-7	74.60	91.89
et-sl	-61594	-194218	2.80	0.27	117767	15563.93	0.08	-11	66.93	82.60
fi-lv	-66730	-11261	5.33	0.34	-72108	-2285.02	0.06	-8	66.00	99.10
fi-nl	-89718	-393774	1.71	0.30	-89284	-3638.18	0.26	-5	55.46	64.44
fi-pt	-110479	-439797	3.60	0.54	-111720	-6939.93	0.22	-7	45.41	59.84
fi-sk	-59295	-46799	3.96	0.26	-182908	-6349.16	0.08	-10	69.96	95.90
fi-sl	-72939	-134593	4.11	0.32	-71835	8022.41	0.11	-13	62.77	86.20
lv-nl	-19634	-354516	-3.52	-0.05	-18654	-318.15	0.20	-2	84.02	65.02
lv-pt	-41716	-402441	-1.46	0.21	-36193	4468.15	0.16	-6	68.80	60.39
lv-sk	7581	-34478	-1.38	-0.08	-123658	-7706.36	0.02	-5	105.99	96.77
lv-sl	-5810	-122602	-1.21	-0.03	1014	7032.05	0.05	-6	95.10	86.99
nl-pt	-20143	-43781	1.86	0.24	-19930	-314.52	-0.04	-1	81.88	92.87
nl-sk	27385	314730	2.09	-0.04	27329	1161.44	-0.19	3	126.14	148.83
nl-sl	13810	230590	2.32	0.03	7637	-2267.56	-0.16	-3	113.18	133.78
pt-sk	48780	361817	0.12	-0.29	47881	1201.90	-0.15	5	154.06	160.25
pt-sl	35260	275981	0.32	-0.22	35831	6622.65	-0.12	0	138.23	144.05
sk-sl	-13637	-85421	0.23	0.07	-130809	-9182.12	0.03	-3	89.73	89.89

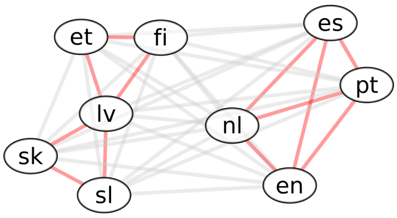


Fig. 2: Language network based on the Clustering coeff. The red links are present after the threshold of 10^{-3} was applied. Gray links represent connections that are not present given the applied threshold. We can see two groups, one formed by Balto-Slavic and Finnic languages, the other by Germanic and Romanic.

6 Discussion and conclusions

In this work, our aim was to provide one of the first large-scale comparisons of languages based on corpus-derived networks. To the best of our knowledge, the use of network topologies on sequence-based token networks are novel and it is not yet known to what characteristics the network topologies correspond. Second, we investigated whether the difference in some metrics correspond known relationships between languages, or represent novel language groupings.

We have shown that the proposed network-based text representation offers a pallete of novel opportunities for language comparison. Commonly, methods operate on sequence level, and are as such limited to one dimensional interactions with respect to a given token. In this work we attempted to lift this constraint by introducing richer, global word neighborhood. We were able to cast the language comparison problem to comparing network topology metrics, for which we show can be informative for genetic and typographic comparisons. For example, the Slovene and Slovak languages appear to have very similar global network structure, indicating comparison using communities picks up some form of evolutionary language distance. In this work we explored only very simple language networks by performing virtually no preprocessing. We believe a similar idea could be used to form networks from lemmatized text or even Universal Dependency Tags, potentially opening another dimension.

Overall, we identified the clustering coefficient as the metric, which, when further inspected, yielded some of the well known language-language relationships, such as for example high similarity between Spanish and Portugese, as well as Slovenian and Slovak languages. Similar observation was made when community structure was compared. We believe such results demonstrate network-based language comparison represents a promising venue for scalable and more informative studies of how languages, and text in general, relate to each other.

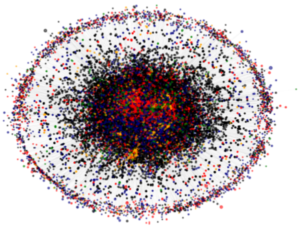


Fig. 3: Visualization of the English DGT subcorpus. This network was constructed using the proposed text2net algorithm, where each link corresponds to the *followed by* relation between a given pair of word tokens. Clustering emerges, indicating the presence of meso-scale topological structures in such networks. Different colors correspond to different communities detected using InfoMap.

In future, we will closer connect the interpretation of network topological features with linguistic properties, also by single language metrics. Also, we believe that document-level classification tasks can benefit from exploiting the inner document structure (e.g., the Graph Aggregator framework could be leveraged instead of/in addition to conventional RNN-based approaches). The added value of graph-based similarity for classification was demonstrated e.g., in [18] for psychosis classification from speech graphs. We also believe that our cross-language analysis, could be indicative for the expected quality of cross-lingual representations. Last but not least, we plan to perform additional experiments to see if the results are stable, leading to similar findings of other corpora genres and corpora of other sizes, and also using comparable not only parallel data.

Acknowledgements The work was supported by the Slovenian Research Agency through a young researcher grant [BŠ], core research programme (P2-0103), and project Terminology and knowledge frames across languages (J6-9372). This work was supported also by the EU Horizon 2020 research and innovation programme, Grant No. 825153, EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media). The results of this publication reflect only the authors views and the EC is not responsible for any use of the information it contains.

References

1. Asgari, E., Schütze, H.: Past, present, future: A computational investigation of the typology of tense in 1000 languages. arXiv:1704.08914 (2017)
2. Bjerva, J., Östling, R., Han Veiga, M., Tiedemann, J., Augenstein, I.: What do language representations really represent? Computational Linguistics (0), 381–389 (2019)

3. Bollobás, B.: Modern graph theory, vol. 184. Springer Science & Business Media (2013)
4. Boudin, F.: pke: an open source python-based keyphrase extraction toolkit. In: Proc. of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations. pp. 69–73. Osaka, Japan (December 2016)
5. Brandao, M.A., Moro, M.M.: Social professional networks: A survey and taxonomy. Computer Communications **100**, 20–31 (2017)
6. Daniel, M.: Linguistic typology and the study of language. pp. 43–68. Oxford University Press (Nov 2010)
7. De Domenico, M., Lancichinetti, A., Arenas, A., Rosvall, M.: Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. Physical Review X **5**(1), 011027 (2015)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
9. Fortunato, S.: Community detection in graphs. Physics reports **486**(3-5), 75–174 (2010)
10. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems **151**, 78–94 (2018)
11. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using networkx. Tech. rep., LANL, Los Alamos, NM, USA (2008)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
13. Kepser, S., Reis, M.: Linguistic evidence: empirical, theoretical and computational perspectives, vol. 85. Walter de Gruyter (2008)
14. Kettunen, K., Sadeniemi, M., Lindh-Knuutila, T., Honkela, T.: Analysis of eu languages through text compression. In: International Conference on Natural Language Processing (in Finland). pp. 99–109. Springer (2006)
15. Kralj, J., Robnik-Sikonja, M., Lavrac, N.: Netsdm: Semantic data mining with network analysis. Journal of Machine Learning Research **20**(32), 1–50 (2019)
16. Loper, E., Bird, S.: Nltk: the natural language toolkit. arXiv cs/0205028 (2002)
17. McKinney, W.: pandas: a foundational python library for data analysis and statistics. Python for High Performance and Scientific Computing **14** (2011)
18. Mota, N.B., Vasconcelos, N.A., Lemos, N., Pieretti, A.C., Kinouchi, O., Cecchi, G.A., Copelli, M., Ribeiro, S.: Speech graphs provide a quantitative measure of thought disorder in psychosis. PloS one **7**(4), e34928 (2012)
19. Nivre, J., De Marneffe, M.C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al.: Universal dependencies v1: A multilingual treebank collection. In: Proc. of the 10th International Conference on Language Resources and Evaluation (LREC 2016). pp. 1659–1666 (2016)
20. Port, A., Gheorghita, I., Guth, D., Clark, J.M., Liang, C., Dasu, S., Marcolli, M.: Persistent topology of syntax. Mathematics in Computer Science **12**(1), 33–50 (2018)
21. Rama, T., Kolachina, P.: How good are typological distances for determining genealogical relationships among languages? Proc. of COLING 2012: Posters pp. 975–984 (2012)
22. Rosvall, M., Axelsson, D., Bergstrom, C.T.: The map equation. The European Physical Journal-Special Topics **178**(1), 13–23 (2009)
23. Škrlj, B., Kralj, J., Lavrač, N.: Py3plex: a library for scalable multilayer network analysis and visualization. In: International Conference on Complex Networks and their Applications. pp. 757–768. Springer (2018)