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

D1.6: Final cross-lingual and multilingual embeddings technology (T1.1)

Executive summary
In this deliverable, we present advances in cross-lingual embeddings technology, developed in T1.1 of the EMBEDDIA project. By aligning vector spaces of several languages, cross-lingual embeddings can be used to transfer machine learning models between languages, thereby compensating for insufficient data in less-resourced languages. We first propose a novel approach to contextual cross-lingual embeddings based on generative adversarial networks (GANs) that drops the assumption of isomorphic spaces in different languages. Second, we test modern cross-lingual transfer approaches on Twitter sentiment prediction task in 13 languages, focusing on the joint numerical space for many languages as implemented in the LASER library and variants of multilingual BERT models. Third, we developed a contextual embeddings-based approach to detect idiomatic expressions, one of the difficult linguistic problems. We successfully applied the solution in monolingual and cross-lingual settings, using ELMo and BERT monolingual and cross-lingual embeddings. Overall, our results show that using modern contextual cross-lingual technologies enables a successful cross-lingual transfer of prediction models to less-resourced languages, achieving almost zero-loss compared to direct training in target languages for some problems.

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:	**D1.6**
Deliverable full title:	**Final cross-lingual and multilingual embeddings technology**
Deliverable short title:	**Final cross-lingual embeddings**
Document identifier:	**EMBEDDIA-D16-FinalCrosslingualEmbeddings-T11-submitted**
Lead partner short name:	**UL**
Report version:	**submitted**
Report submission date:	**31/12/2020**
Dissemination level:	**PU**
Nature:	**R = Report**
Lead author(s):	**Marko Robnik-Šikonja (UL), Matej Ulčar (UL), Luka Krsnik (UL)**
Co-author(s):	**Tadej Škvorc (JSI), Igor Mozetič (JSI), Luis Adrián Cabrera Diego (ULR), Kristjan Reba (UL)**
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
11/11/2020	v1.0	Marko Robnik-Šikonja (UL)	First draft.
15/11/2020	v1.1	Matej Ulčar (UL), Luka Krsnik (UL)	Contributing contents
02/12/2020	v1.2	Marko Robnik-Šikonja (UL), Matej Ulčar (UL), Luka Krnik (UL))	Forming the report.
09/12/2020	v1.3	Antoine Doucet (ULR)	Internal review.
11/12/2020	v1.4	Hannu Toivonen (UH)	Internal review.
17/12/2020	v1.5	Marko Robnik-Šikonja (UL), Matej Ulčar (UL)	Revision based on internal reviews.
19/12/2020	v1.6	Nada Lavrač (JSI)	Report quality checked and finalised.
20/12/2020	v1.7	Marko Robnik-Šikonja (UL)	Final corrections.
29/12/2020	submitted	Tina Anžič (JSI)	Report submitted.

Table of Contents

1. Introduction 5

2. Cross-lingual mappings of contextual text embeddings 7

2.1 Background and related work on cross-lingual embeddings 7
2.1.1 Alignment of monolingual embeddings 7
2.1.2 ELMo contextual embeddings 8

2.2 Improvements in contextual cross-lingual mapping approaches 9
2.2.1 Novel cross-lingual contextual dataset 9
2.2.2 Novel contextual cross-lingual isomorphic mappings 10
2.2.3 Novel contextual cross-lingual non-isomorphic mapping with GANs 11
2.2.4 Mapping with additional anchor points 12

2.3 Cross-lingual mappings evaluation and results 13
2.3.1 Named entity recognition 14
2.3.2 Dependency parsing 17

2.4 Discussion on cross-lingual contextual mappings 20

3. Cross-lingual model transfer for sentiment prediction 21

3.1 Cross-lingual transfer technologies 22
3.1.1 Projecting into a common vector space 22
3.1.2 BERT contextual model 22

3.2 Datasets and experimental settings 23
3.2.1 Evaluation metrics 23
3.2.2 Datasets 23
3.2.3 Implementation details 24

3.3 Experiments in cross-lingual transfer 24

3.4 Comparing representations 26

3.5 Discussion on cross-lingual sentiment predictors transfer 26

4. Cross-lingual transfer for prediction of idioms 27

4.1 Novel MICE architecture for idiom detection 28

4.2 Datasets of idiomatic expressions 29
4.2.1 Novel monolingual dataset 29
4.2.2 PARSEME datasets 30

4.3 Evaluation 31
4.3.1 IEs from the training set 32
4.3.2 IEs outside the training set 34
4.3.3 Cross-lingual evaluation of IEs 34

4.4 Discussion on idiom detection 35

5. Conclusions and further work 37

6. Associated outputs 37

References 39

Appendix A: Cross-lingual alignments of ELMo contextual embedding 42

Appendix B: Cross-lingual Transfer of Sentiment Predictors 66

Appendix C: MICE: Mining Idioms with Contextual Embeddings 82

List of abbreviations

BERT	Bidirectional Encoder Representations from Transformers
CBOW	Continuous Bag Of Words
CNN	Convolutional Neural Network
CSLS	Cross-domain Similarity Local Scaling
ELMo	Embeddings from Language Models
GAN	Generative Adversarial Networks
IE	Idiomatic Expression
LSTM	Long Short-term Memory
MLM	Masked Language Model
MWE	Multi-Word Expression
NLP	Natural Language Processing
NE	Named Entity
NEL	Named Entity Linking
NER	Named Entity Recognition

1 Introduction

The EMBEDDIA project aims to improve the cross-lingual transfer of language resources and trained models using word embeddings and cross-lingual word embeddings. 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. The results of this WP form a technological basis for other WPs in the project, in particular WP3, WP4, and WP5 that work on concrete news media problems. To demonstrate advancements, EMBEDDIA covers English and eight less-resourced languages: Croatian, Estonian, Finnish, Latvian, Lithuanian, Russian, Slovene, and Swedish. The specific objectives of WP1 are as follows:

1. advance cross-lingual and multilingual word embeddings technology in T1.1,
2. advance context-dependent and dynamic embeddings technology in T1.2,
3. advance deep learning technology for morphologically rich, less-resourced languages in T1.3,
4. improve the interpretability of models and visualisation of results in T1.4,
5. collect and prepare datasets and benchmarks required to evaluate the developed technologies in T1.5.

The difference between T1.1 and T1.2 is that T1.1 focuses on cross-lingual embeddings and cross-lingual transfer of models, while T1.2 develops novel contextual and dynamic embeddings and novel approaches for their evaluation.

This report describes the results of the work performed in T1.1 from M13 to M24 and is the final deliverable of this task. The initial work within T1.1 from M1 to M12 was reported and accepted as deliverable D1.2 in M12. That work covered cross-lingual alignment of non-contextual embedding, while in this report, we focus on cross-lingual alignment of contextual embeddings and the analysis of cross-lingual model transfer in different areas.

Word embeddings are representations of words in numerical form, as vectors of typically several hundred dimensions. The vectors are used as input to machine learning models; these are generally deep neural networks for complex language processing tasks. The embedding vectors are obtained from specialized neural network-based embedding algorithms, e.g., word2vec (Mikolov et al., 2013) or fastText (Bojanowski et al., 2017), or more recent contextual approaches such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019).

Modern word embedding spaces exhibit similar structures across languages, even when considering distant language pairs like English and Vietnamese (Mikolov et al., 2013). This means that embeddings independently produced from monolingual text resources can be aligned, resulting in a common cross-lingual representation, called cross-lingual embeddings, which allows for fast and effective integration of information in different languages. Cross-lingual approaches can be sorted into several groups. The first group of methods uses monolingual embeddings with (an optional) help from bilingual dictionaries to align the embeddings. The second group of approaches uses bilingually aligned (comparable or even parallel) corpora for joint construction of embeddings in all involved languages. The third type of approach is based on large pretrained multilingual masked language models such as BERT (Devlin et al., 2019). The multilingual BERT is typically used as a starting model, which is fine-tuned for a particular task without explicitly extracting embedding vectors.

While in previous work, reported in D1.2, we analyzed the most successful approaches to *non-contextual* cross-lingual embeddings, in this report, we focus on improvements of cross-lingual mappings for modern *contextual* embeddings. Currently, the most successful approaches to cross-lingual mappings assume that the embedding spaces in different languages are isomorphic (Artetxe et al., 2018b; Conneau et al., 2018), which is generally not the case. Several researchers have observed that the monolingual embedding spaces of two different languages are not completely isomorphic, which is especially true for distant languages (Ormazabal et al., 2019; Søgaard et al., 2018). As a result, many of these methods are unstable or unsuccessful when confronted with distant language pairs. We propose novel methods

for isomorphic and non-isomorphic alignment of contextual embeddings, such as ELMo. For that purpose, we first construct novel contextual mapping datasets based on parallel corpora and dictionaries. In a novel non-isomorphic approach, we use generative adversarial networks (GANs) (Goodfellow et al., 2014), that produce non-linear mappings between the embedding spaces. Additionally, we test several different types of anchor points between languages, such as low- and high-quality dictionaries, named entities, and linked entities obtained from multilingual lexicalized semantic network BabelNet (Navigli & Ponzetto, 2012). As contextual embeddings such as ELMo became the essential NLP technologies, their successful cross-lingual mappings, developed in T1.1, enable the cross-lingual lingual transfer of text enrichment and keyword extraction technologies developed in WP2, transfer of comment filtering models developed in WP3, and sentiment classification models developed in WP4.

As the second important aspect of cross-lingual technologies, we analyze the practical issue of cross-lingual transfer of trained machine learning prediction models between languages. Using large datasets of Twitter sentiment in 13 languages, we compare two practically important cross-lingual model transfer approaches. The first one is based on mappings of words into a common vector space, as implemented in the LASER library (Artetxe & Schwenk, 2019). The results show that there is a significant transfer potential using the models trained on similar languages. The second approach to model transfer is multilingual BERT model (Devlin et al., 2019), trained simultaneously on 104 languages, and its variants trained on only three languages (Ulčar & Robnik-Šikonja, 2020). The results show that the variants of multilingual BERT with fewer languages are the most successful. This work is relevant for WP3 and WP4, as it shows the advantages and limits of cross-lingual models developed there, e.g., cross-lingual opinion analysis in T3.1 and multilingual news linking in T4.1.

As the third contribution, we demonstrate the practical use of modern cross-lingual technologies, particularly cross-lingual use of contextual embeddings. As a testbed, we use an important problem from linguistics, namely the detection of idiomatic expressions. Good coverage and accuracy of idiom detection and identification tools are important components of many NLP applications, such as word sense disambiguation and machine translation. We show that deep neural networks using either ELMo or BERT embeddings (produced in T1.2) perform much better than existing approaches and can detect idiomatic word use even for idioms that were not present in the training set. We demonstrate the cross-lingual transfer of developed models and show that contextual word embeddings can generalize to other languages. This work is relevant for T1.2, as it demonstrates the cross-lingual aspect of monolingual contextual embeddings (ELMo and BERT) developed there. Further, as idioms are an important aspect of creative language use, our approach can improve the work in T5.3 on creative language use for multilingual news and headline generation.

The main contributions presented in this report (in the order of appearance) are as follows.

1. Development of a novel dataset and methods for cross-lingual alignment of contextual embeddings based on isomorphic and non-isomorphic transformations, presented in Section 2 and the paper by Ulčar & Robnik-Šikonja (2020), included in Appendix A.
2. Analysis of the cross-lingual transfer of prediction models, using Twitter sentiment prediction as a use case, described in Section 3 and the paper by Robnik-Šikonja et al. (2020), included in Appendix B.
3. Development of a novel contextual-embedding approach for detection of idiomatic expressions, showing superior performance in monolingual and multilingual settings, as well as in cross-lingual transfer, described in Section 4 and the paper by Škvorc et al. (2020), included in Appendix C.

Besides these contributions, the work in T1.1 has contributed to achievements reported in other work-packages, in particular to improvements in contextual embeddings (T1.2), semantic enrichment (T2.1), keyword extraction (T2.2), context and opinion analysis (T3.1), comment filtering (T3.2), cross-lingual identification of sentiment (T4.3), and creative language use (T5.3). The produced resources are being integrated into EMBEDDIA Media Assistant and ClowdFlows platform as contributions to WP6 and WP7, respectively. The availability of new resources produced in this work is discussed in Section 6.

2 Cross-lingual mappings of contextual text embeddings

This section reports on the work done in T1.1 on improvements in cross-lingual contextual mappings between different languages. In deliverable D1.2, we already reported on the quality of different *non-contextual* cross-lingual mappings. In this work, we propose several improvements to *contextual* cross-lingual mappings. We first give background and overview of different approaches to cross-lingual embeddings in Section 2.1. In Section 2.2, we describe the improvements to cross-lingual mappings of contextual embeddings and evaluate them in Section 2.3. We discuss the implications of the results in Section 2.4.

2.1 Background and related work on 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.

There are three groups of approaches to find cross-lingual mappings. The first group of approaches, presented in Section 2.1.1, uses monolingual embeddings with the optional help from a bilingual dictionary to align the embeddings. In this section, we only work with this group of cross-lingual approaches, while the remaining two groups are analyzed in Section 3. The second group of approaches uses bilingually aligned (comparable or even parallel) corpora for joint construction of embeddings in all involved languages. The third type of approaches are based on large pretrained multilingual masked language models such as BERT (Devlin et al., 2019). The multilingual BERT is typically used as a starting model, which is fine-tuned for a particular task without explicitly extracting embedding vectors. For this reason, our work on novel cross-lingual mappings is focused on the ELMo contextual embeddings, presented in Section 2.1.2. At the same time, we use multilingual BERT models in Section 3 and Section 4.

2.1.1 Alignment of monolingual embeddings

Cross-lingual alignment methods take precomputed word embeddings for each language and align them with the 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 other language's vector space (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 works by Artetxe et al. (2018a).

The open source implementation of the method described by Artetxe et al. (2018b,a), named *vecmap*[1], is able to align monolingual embeddings either using supervised, semi-supervised or unsupervised approach.

The supervised approach requires using a large bilingual dictionary, which is used to match embeddings of the 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).

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

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 the 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 help as its entries can be used as anchors for the alignment map for supervised and semi-supervised approaches. However, lately, researchers have proposed approaches that do not require a bilingual dictionary but rely on an adversarial approach (Conneau et al., 2018) or use the frequencies of the words (Artetxe et al., 2018b) to find a required transformation. These are called unsupervised approaches.

The Facebook research project MUSE[2] can find a cross-lingual map with the use of a bilingual dictionary (supervised) or without one (unsupervised approach). The unsupervised approach works by using adversarial training to find the starting linear mapping. A synthetic dictionary is extracted from this mapping, which is used to fine-tune the starting mapping using the Procrustes approach, described in detail by Conneau et al. (2018).

2.1.2 ELMo contextual embeddings

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 (Hochreiter & Schmidhuber, 1997) . 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, one puts one sentence at a time on the input. 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. Still, 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 the 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). Later, BERT models turned out to be even more successful on these tasks. However, concerning the quality of extracted vectors, ELMo is often advantageous. The information it contains is condensed

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

in only three layers, while multilingual BERT uses 14 layers. This observation is confirmed in our experiments, reported in Section 4. For that reason, we develop cross-lingual mappings suitable for ELMo precomputed contextual models in Section 2.2. The actual implementations of ELMo models come from our work in T1.2 (Ulčar & Robnik-Šikonja, 2020a), where we developed modern contextual embeddings approaches (ELMo and BERT) for the languages covered in the EMBEDDIA project.

2.2 Improvements in contextual cross-lingual mapping approaches

Context-dependent models calculate a word embedding for each word's occurrence; 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 the 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 appear in 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 that take different contexts and word meanings into account. Both methods require a different type of contextual mapping datasets that maps not only words but also their contexts. We describe the creation of such novel datasets in Section 2.2.1. The first novel contextual mapping approach, described in Section 2.2.2, requires these datasets but still uses the monolingual embedding mapping methods (described in Section 2.1.1) for alignment of contextual embeddings. The second cross-lingual contextual mapping approach, described in Section 2.2.3, uses the same contextual datasets but drops the assumption that the aligned spaces are isomorphic; it uses GANs to form a non-linear transformation between contextual embeddings. The quality of cross-lingual mappings depends on the quality and size of dictionaries used to create contextual cross-lingual mapping datasets. High-quality dictionaries are nonexistent or not freely available for many language pairs, especially for low resourced languages. For that reason, in Section 2.2.4, we present how other sources of potentially useful alignment points, such as linked named entities and multilingual semantic network BabelNet, can be employed in contextual cross-lingual mappings.

2.2.1 Novel cross-lingual contextual dataset

The main obstacle to form a cross-lingual mapping between contextual embeddings is that a word in one language is represented with several different words (one for each meaning) in another language. We propose two novel methods for the 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).

We used the OpenSubtitles parallel corpus[3] (Lison & Tiedemann, 2016) from the Opus web page[4] for each pair of languages that we evaluated. The dictionaries we used are bilingual dictionaries extracted

[3] https://www.opensubtitles.org/.
[4] http://opus.nlpl.eu

from wiktionary, using wikt2dict[5] tool (Acs, 2014). We extracted dictionaries for each EMBEDDIA language paired with English and the following language pairs of similar languages: Croatian-Slovenian, Estonian-Finnish, and Latvian-Lithuanian. For the language pairs not involving English, we created two different dictionaries, a direct bilingual dictionary and a dictionary created with triangulation via English. Dictionaries created with triangulation have more entries but are of worse quality than direct dictionaries. After the extraction, we manually cleaned the dictionaries using filters, such as removing accent marks on vowels from languages that do not use them (e.g., Slovenian) and removing extra non-alphabetical characters, like brackets, colon, and hash. We limited the dictionaries to entries with single-word terms in both languages. The sizes of the obtained dictionaries and parallel corpora used are displayed in Table 1. Additionally, to test the impact the quality and size of a dictionary have, we used a proprietary high-quality human-made Oxford English-Slovene dictionary for that pair of languages.

Table 1: The sizes of dictionaries and parallel corpora used in the creation of a dataset for contextual mappings, as well as the size of the resulting dataset for alignment of ELMo embeddings (see Section 2.2.2). The languages are represented with their international language codes ISO 639-1. The sizes of dictionaries are reported in the number of pairs, the sizes of parallel corpora in the number of matching contexts, and the sizes of resulting datasets in the number of matched words in matched sentence pairs. The Type column describes the dictionary creation approach: "direct" means that the dictionary was created directly from wiktionary, "triang" means that the dictionary was created from wiktionary using triangulation via English, and "OES" stands for the Oxford English-Slovene dictionary.

Language pair	Type	Bilingual dictionary	Parallel corpus	ELMo dataset
en-et	direct	11 022	12 486 898	77 800
en-fi	direct	89 307	27 281 566	283 000
en-hr	direct	3448	35 131 729	44 800
en-lt	direct	13 960	1 415 961	62 800
en-lv	direct	10 224	519 553	43 800
en-ru	direct	103 850	25 910 105	363 800
en-sl	direct	9634	19 641 457	89 800
en-sl	OES	182 787	19 641 457	294 318
en-sv	direct	51 961	17 660 152	270 000
et-fi	direct	2191	9 504 879	12 800
et-fi	triang	43 313	9 504 879	78 200
hr-sl	direct	266	15 636 933	3400
hr-sl	triang	3669	15 636 933	31 600
lt-lv	direct	2478	219 617	11 200
lt-lv	triang	14 545	219 617	28 200

2.2.2 Novel contextual cross-lingual isomorphic mappings

The first method we propose for computation of cross-lingual mappings between contextual embeddings is still based on the assumption that the aligned spaces are largely isomorphic. With a large enough collection of words in matching contexts (as described above in Section 2.2.1), we compute their contextual embedding vectors and align them with any of the non-contextual mapping methods, either with vecmap library (Artetxe et al., 2018a), which showed the best performance in our experiments, reported in deliverable D1.2, or MUSE library (Conneau et al., 2018), which only aligns target vectors and is therefore computationally more efficient as discussed later. To test this approach, we work with ELMo contextual embeddings due to their advantage over BERT concerning extracted vectors, as explained in Section 2.1. The ELMo model is shortly presented in Section 2.1.2.

Recently, a similar approach was proposed by Aldarmaki & Diab (2019) but did not use large contextual datasets based on high-quality dictionaries 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

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

(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.

In our isomorphic method for alignment of ELMo contextual embeddings, called ELMoVM (ELMo with VecMap) or ELMoMU (ELMo with MUSE), we approached the creation of the contextual mapping dataset in two ways, one for contextual ELMo layers and the second for the non-contextual ELMo layer. For contextual ELMo layers, we lemmatized the parallel corpora using the Stanza tool (Qi et al., 2020). We then processed each corpus context by context. For each context, we calculated the embeddings of the non-lemmatized corpus. We then checked for each word of the lemmatized context if its pair from the bilingual dictionary appears in the same lemmatized context of the other language. When such a match was found, the two words' IDs and their ELMo embeddings were added to the list of anchors. The reason for the lemmatization is that the bilingual dictionaries predominantly contain lemmas of the words. Note that we still use the non-lemmatized corpus in the computation of embeddings to get the correct contexts. In creating the contextual mapping dataset, we considered at most 20 different contexts of each lemma to not overwhelm the dataset with frequent words (such as stop words). The size of the created dataset is displayed in the right-most column of Table 1.

The first of the three ELMo layers is non-contextual, so we used a different approach for vectors from that layer. We first calculated embeddings for each pair of words in the bilingual dictionary. We used that as our list of anchors. We split the created datasets of anchor lists into the training and testing part. The training part takes 98.5% of the whole dataset for each language pair, and the testing part takes 1.5%. These datasets were used to map one vector space to another, allowing us to map one word with multiple meanings in one language to multiple words in another language.

We used the computed bilingually aligned contextual embedding pairs as an input to methods that align two monolingual embeddings (Section 2.1.1). To get the cross-lingual alignment, we used the vecmap supervised method (Artetxe et al., 2018a) or MUSE supervised method (Conneau et al., 2018).

2.2.3 Novel contextual cross-lingual non-isomorphic mapping with GANs

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 commonly used vecmap and MUSE methods.

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are a type of neural networks consisting of two connected neural models, a generator and a discriminator. The two models are trained simultaneously via an adversarial process. The discriminator attempts to discern whether the data passed to its input is real or fake (i.e. artificially generated). At the same time, the generator attempts to generate artificial data, which can fool the discriminator. GANs play a zero-sum game, where a success of the discriminator means a failure of the generator and vice versa. By simultaneously training both networks, they both improve. GANs are mostly used on images, where the described process can lead to compelling new generated images.

Following the success of GANs in neural machine translation (Yang et al., 2018) and unsupervised cross-lingual alignment (Conneau et al., 2018; Fu et al., 2020), we propose a novel supervised non-linear mapping method using bidirectional GANs. We based our contextual alignment method, called ELMo-GAN, on the model of Fu et al. (2020). Contrary to Fu et al. (2020), who only used their method with non-contextual fastText embeddings (Bojanowski et al., 2017) to align sentences, we align contextual ELMo embeddings (Peters et al., 2018), which is only possible by constructing a special contextual mapping datasets, described in Section 2.2.1.

As illustrated in Figure 1, the mapping GAN comprises the generator module and discriminator module. The generator module contains two generators that map vectors from one vector space to the other. Specifically, for a pair of languages L_1 and L_2, one generator will map from L_1 to L_2, and the second will map from L_2 to L_1. Discriminator module contains two discriminators. The first discriminator tries to predict whether a given pair of vectors represent the same token, i.e. if the first vector represents the word x in L_1 and the second vector represents the translation of the word x in L_2. The second discriminator attempts to learn the difference between the direction of mapping. For a given pair of vectors, it predicts whether they are a vector from L_1 and its mapping to L_2 or a vector from L_2 and its mapping to L_1.

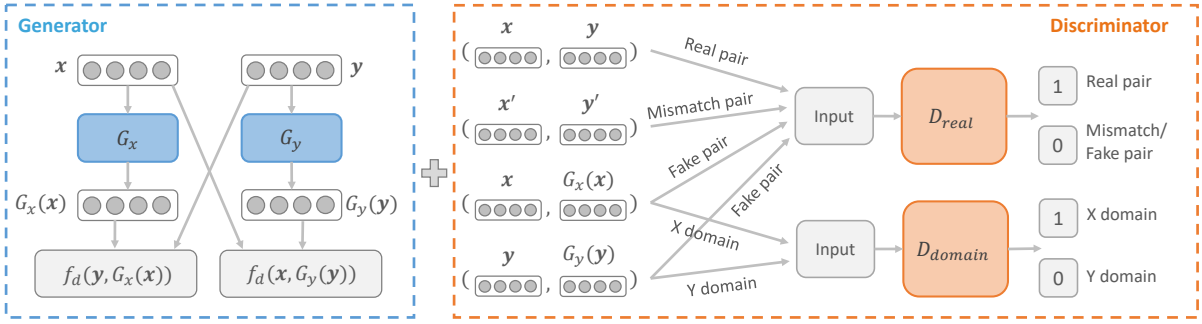


Figure 1: The schema of the GAN, proposed by Fu et al. (2020) for sentence alignment. The image is taken from that source.

Compared to the ABSent model by Fu et al. (2020), in ELMo-GAN, we increased the sizes of all the hidden layers in generators and discriminators. We also significantly lowered the learning rate as we achieved poor results with the learning rate used by (Fu et al., 2020). Both generators have the same architecture: the input layer is followed by three fully connected feed-forward layers of sizes 2048, 4096, and 2048. We used the ReLU activation function for all three layers and added a batch normalization layer between each fully connected layer. The output layer has the same size as the input layer. It uses hyperbolic tangent as the activation function so that the output is between -1 and $+1$. Both discriminators also have the same architecture. We first concatenate the two input vectors, then feed them to three consecutive fully connected feed-forward layers with leaky ReLU ($\alpha = 0.2$). The output layer is a single neuron with the sigmoid activation.

We jointly trained the generator and discriminator modules using the parallel ELMo vectors datasets, described in Section 2.2.1. We trained ELMoGAN with the batch size of 256, Adam optimizer with learning rate 2×10^{-5}, and learning rate decay 10^{-5}. For each language pair, we trained three mapping models, one for each of the ELMo layers. For all three models, we used the same settings.

We produced two different versions of the ELMoGAN, based on the number of iterations the model was trained for. The first version (ELMoGAN-10k) was trained for a fixed number of 10 000 iterations for each layer of each language pair. The second version (ELMoGAN-O) was trained for the number of iterations that gave the best result in the dictionary induction task, using the evaluation dictionary from Section 2.2.1. This choice was determined in our preliminary tests on unrelated NER tasks. As our experiments on the dependency parsing task show, these choices might not be general enough and require different settings and evaluation tasks for other problems.

2.2.4 Mapping with additional anchor points

Anchors, connecting identical points in cross-lingual mappings between two languages, are selected from bilingual dictionaries. The majority of high-quality bilingual dictionaries are proprietary and, in some cases, do not exist at all, especially for less-resourced languages and geographically distant regions. We explored several alternatives to dictionaries, which can be used as anchor points in cross-lingual mappings: named entities (NE) obtained via named entity recognition (NER) in collaboration with

T2.1, and connections between different entities contained in the BabelNet semantic network. Ideally, we could also obtain anchor points from the named entity linking (NEL) task. Unfortunately, our initial attempts to obtain reasonably good anchor points for EMBEDDIA languages with NEL in collaboration with T2.1 did not produce satisfactory results, so we leave this research direction for further work. Below we describe NE anchors obtained from NER and BabelNet.

NER prediction models exist for many languages, as the NER task is one of the standard NLP tasks. To obtain anchor points for cross-lingual alignment of contextual embeddings, we must find matching NEs in the same context. Following the methodology described in Section 2.2.1, we attempt to do so via parallel corpora. There exist multiple aligned parallel multilingual corpora; in our experiments we used MultiParaCraawl and ParaCrawl[6] corpora available on the Opus website[7]. In collaboration with T2.1, we used NER algorithms, described in delivery D2.5, to annotate the parallel corpora in different languages. We annotate each language in a bilingual pair with its corresponding monolingual NER prediction model. To obtain anchor points, we match named entities between the languages in the given pair on a sentence level. If matching sentences in both languages contain a single NE with the same label, we consider them to be translations of one another and store them as anchor points. The dictionary we form contains NEs matched in this way.

BabelNet is a semantic network connecting concepts and named entities (Navigli & Ponzetto, 2012) in 284 different languages. BabelNet was created by automatically extracting connections from multiple sources such as WordNet, Wikipedia, OmegaWiki, Wiktionary, etc. We use BabelNet connections to extract bilingual translations of different entities and form a dictionary of anchoring points from them.

Manual inspection shows that NER and BabelNet dictionaries of anchor points contain many errors. We believe their quality is inappropriate for human usage. Still, we test if the information they contain is of sufficient quality to improve cross-lingual alignment. We compare the sizes of different dictionaries, corpora sizes for each language pair, and sizes of the resulting contextual dataset for ELMo embeddings' alignment in Table 2. To give a reference point, besides the automatically harvested dictionaries, we also present the size of human-made proprietary Oxford English-Slovene dictionary (OES).

Table 2: The sizes of anchor point dictionaries, number of sentences in parallel corpora, and sizes of the resulting contextual datasets for alignment of ELMo embeddings (see Section 2.2.2 for the methodology). The Type column labels the type of dictionary: BNd stands for BabelNet dictionary, NERd for dictionary extracted with NER from parallel corpora, and OES for human-made Oxford English-Slovene dictionary.

Language pair	Type	Bilingual dictionary	Parallel corpus	ELMo dataset
hr-sl	BNd	80 309	271 415	125 600
hr-sl	NERd	24 708	271 415	32 000
en-hr	BNd	117 682	1 861 590	226 000
en-hr	NERd	22 480	1 861 590	39 400
en-sl	OES	182 786	1 406 645	275 000
en-sl	BNd	105 974	1 406 645	222 400
en-sl	NERd	15 844	1 406 645	27 800

2.3 Cross-lingual mappings evaluation and results

We evaluated the proposed cross-lingual embedding approaches on two downstream NLP tasks: named entity recognition (NER) and dependency parsing (DP). We report results separately for each of the two tasks, NER in Section 2.3.1 and DP in Section 2.3.2.

We use three evaluation settings for each downstream task: comparison of novel contextual mapping techniques, the impact of dictionary type, and the analysis of the dictionary size concerning the quality of embeddings.

[6] https://paracrawl.eu/
[7] http://opus.nlpl.eu/

In the comparison of novel contextual mapping techniques, we evaluate the new cross-lingual mapping methods against three baselines: two existing mapping methods, vecmap and MUSE, and direct learning on target language without cross-lingual transfer. In the dictionary type analysis setting, we compare our new augmented dictionaries against one baseline: direct learning on target language without cross-lingual transfer.

As contextual embeddings, we use ELMo embeddings (Peters et al., 2018), computed in T1.2 for EMBEDDIA languages (Ulčar & Robnik-Šikonja, 2020a). We did not include explicit BERT vectors in this comparison as multilingual BERT is inherently multilingual and does not need an explicit alignment. The vectors we extract from BERT are of lower quality than ELMo, as our experiments in Section 4 show.

2.3.1 Named entity recognition

NER is an information extraction task that seeks to locate and classify NE 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 four labels present in all the addressed working languages. These labels are person, location, organization, and other. The Other label encompasses all named entities that do not fall in one of the three mentioned classes, as well as all the tokens that are not named entities. The datasets used in the NER task in this report are shown in Table 3 and described in detail in deliverable D1.1. These datasets are different from the standard NER task (e.g., used in WP2 and deliverables D2.2 and D2.5). We are only interested in NER for comparing different alignment methods and not to maximally improve the NER performance (e.g., we do not use any external information, fine-tuning of models, etc.).

Table 3: The collected datasets for NER task and their properties: number of sentences, number of tagged words, availability, and link to the corpus location).

Language	Corpus	Sentences	Tags	Avail.	Location
Croatian	hr500k	25000	29000	public	link
English	CoNLL-2003 NER	21000	44000	public	link
Estonian	Estonian NER corpus	14000	21000	public	link
Finnish	FiNER data	14500	17000	public	link
Latvian	LV Tagger train data	10000	11500	public	link
Lithuanian	TildeNER	5476	7024	limited	NA
Russian	factRuEval-2016	5000	9500	public	link
Slovene[8]	ssj500k	9500	9500	public	link
Swedish	Swedish NER	8500	7500	public	link

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 Other (i.e. not a named entity).

We use three evaluation scenarios: comparison of novel contextual cross-lingual mapping techniques, the impact of dictionary type, and the impact of the dictionary size on the quality of embeddings. We report specific settings and results for each of the scenarios below.

Comparison of novel contextual cross-lingual mapping techniques

We compare two types of methods for cross-lingual mappings of contextual embeddings. The ELMoVM and ELMoMU methods, presented in Section 2.2.2, assume the existence of isomorphic translation

[8]The Slovene ssj500k originally contains more sentences, but only 9500 are annotated with NER data.

between mapping spaces and use vecmap and MUSE linear translators, respectively, to compute the mapping between two ELMo monolingual embeddings. The ELMoGAN method, presented in 2.2.3, does not assume isomorphism and uses non-linear transformation using GANs. We test two variants of ELMoGAN, the first one, called ELMOGAN-10k, was trained for 10 000 iterations for each layer of each language pair. The second version, called ELMoGAN-O, was trained for the number of iterations that gave the best result in the dictionary induction task, using the evaluation dictionary from Section 2.2.1.

For each method and each of the tested languages, we construct a prediction model for the NER task in the source language. The prediction model is then evaluated on the target language dataset, using target language ELMo embeddings, mapped to the source language vector space. We report the results in Table 4.

Table 4: Comparison of different methods for cross-lingual mapping of contextual ELMo embeddings, evaluated on the NER task. The best result (Macro F1 score) for each language pair is in bold. The reference score "reference" represents a direct learning on the target language without cross-lingual transfer.

Source.	Target.	Dictionary	ELMoVM	ELMoGAN-O	ELMoGAN-10k	ELMoMU	Reference
English	Croatian	direct	**0.385**	0.279	0.345	0.024	0.810
English	Estonian	direct	0.554	0.682	**0.737**	0.284	0.895
English	Finnish	direct	0.672	0.708	**0.788**	0.229	0.922
English	Latvian	direct	0.499	**0.650**	0.630	0.216	0.818
English	Lithuanian	direct	0.498	0.476	**0.575**	0.208	0.755
English	Slovenian	direct	0.548	0.588	**0.664**	0.060	0.850
English	Swedish	direct	0.786	0.686	**0.797**	0.568	0.852
Croatian	Slovenian	direct	0.387	0.279	0.250	**0.418**	0.850
Croatian	Slovenian	triangular	**0.731**	0.365	0.420	0.592	0.850
Estonian	Finnish	direct	**0.517**	0.288	0.302	0.278	0.922
Estonian	Finnish	triangular	**0.779**	0.705	0.677	0.296	0.922
Finnish	Estonian	direct	0.477	0.263	0.331	**0.506**	0.895
Finnish	Estonian	triangular	0.581	0.563	**0.595**	0.549	0.895
Latvian	Lithuanian	direct	**0.423**	0.376	0.367	0.345	0.755
Latvian	Lithuanian	triangular	0.569	0.632	**0.637**	0.378	0.755
Lithuanian	Latvian	direct	0.263	0.305	0.318	**0.604**	0.818
Lithuanian	Latvian	triangular	0.359	0.691	**0.713**	0.710	0.818
Slovenian	Croatian	direct	0.361	0.260	0.328	**0.485**	0.810
Slovenian	Croatian	triangular	**0.566**	0.490	0.427	0.518	0.810

The upper part of Table 4 shows a typical cross-lingual transfer learning scenario, where the model is transferred from resource-rich language (English) to less-resourced languages. In this case, the non-isomorphic ELMoGAN methods, in particular the ELMoGAN-10k variant, are superior to isomorphic ELMoVM and ELMoMU approaches. In this scenario, ELMoGAN-10k is always the best or close to the best mapping approach. This is not always the case in the lower part of Table 4, which shows the second most important cross-lingual transfer scenario: transfer between similar languages. In this scenario, ELMoGAN is the best in three language pairs. Isomorphic ELMoVM and ELMoMU perform best in nine language pairs (5 times ELMoVM and four times ELMoMU). We hypothesize that the reason for the better performance of isomorphic mappings is the similarity of tested language pairs and less violation of isomorphism assumption the ELMoVM and ELMoMU method make. The results of the ELMoMU method support this hypothesis. While ELMoMU performs worst in most cases of transfer from English, the performance gap is smaller for transfer between similar languages. For similar languages, ELMoMU is sometimes the best method, but the results of ELMoMU fluctuate greatly between language pairs. The second possible factor explaining the results is the quality of the dictionaries, which are in general better for combinations involving English. In particular, dictionaries obtained by triangulation via English are of poor quality, and non-isomorphic translation might be more affected by imprecise anchor points.

In general, even the best cross-lingual prediction models lag behind the reference model without cross-lingual transfer. The differences in Macro F_1 score are small for some languages (e.g., 5.5% for English-

Swedish), but they are significantly larger for most of the languages. Compared to cross-lingual transfer with variants of multilingual BERT, described in Section 3, this indicates that ELMo, while useful for explicit extraction of embedding vectors, is less competitive with BERT in the model transfer, especially if we take into account that it requires additional effort for preparation of contextual mapping datasets.

Using NER and BabelNet as anchor points

We analyze how different anchor points impact the quality of cross-lingual mappings. We use two sources of anchor points: NER and BabelNet. To analyze the quality of cross-lingual contextual mappings, we use ELMo-VM mappings, i.e. standard supervised non-isomorphic mappings provided by vecmap library applied to ELMo embeddings. We use the MultiParaCrawl and ParaCrawl corpora to extract matching contextual sentences for ELMo alignment, instead of OpenSubtitles as in the experiment above. The reason for this is that we use NEs as anchor points, and in previous experiments, we noticed that the OpenSubtitles corpus does not have enough realistic NEs with the organization label. While this was not an issue in the previous experiment, it is of significant importance when these points are to become anchor points.

We use the following dictionaries, described in Section 2.2.4: high-quality human-made Oxford English-Slovene dictionary (OES), available only for the English - Slovene language pair, BabelNet dictionary extracted from BabelNet (BNd), and NER dictionary extracted from MultiParaCrawl and ParaCrawl corpora (NERd). The results are contained in Table 5.

In this experiment, we compare all combinations of three languages: Croatian (hr), Slovene (sl), and English (en). We performed the cross-lingual transfer for the NER task for each given language pair L_1-L_2 by training the NER model on L_1 train data and evaluating it on L_2 test data. We compared mapping methods with a reference value obtained via direct learning (direct) and the default value (default) for all three dictionaries. The reference values allow comparison with models without cross-lingual transfer, i.e. training and testing on instances from the same language, without mapping. The default F_1 value is obtained using the majority classifier for each label in the Macro F_1 score for each target language.

Table 5: Impact of different dictionaries as sources of anchoring points on the NER task, measured with the macro F_1 score. We use ELMoVM mapping method. The best non-reference result for each language pair is in bold. OES stands for human made Oxford English-Slovene dictionary, BNd for BabelNet dictionary, NERd for dictionary extracted from NER applied to parallel corpora, “direct“ for direct learning on the target language without cross-lingual transfer, and “default“ for the default F_1 value calculated on the target language.

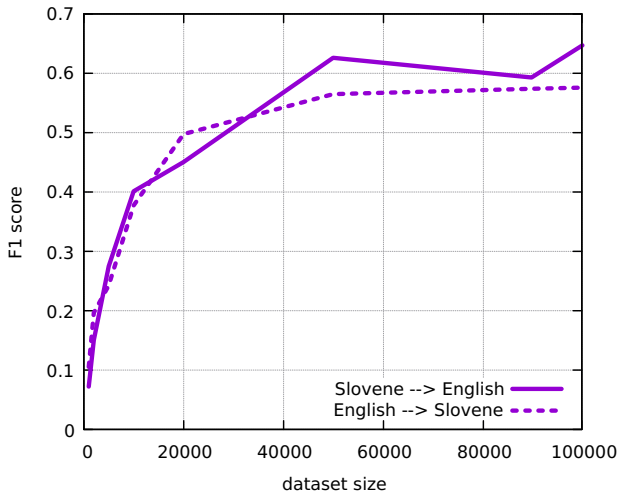
Mapped pair	Train language	Eval language	OES	BNd	NERd	direct	default
hr-sl	hr	sl	/	**0.8150**	0.7545	0.8387	0.0303
hr-sl	sl	hr	/	**0.6187**	0.5668	0.8181	0.0353
en-hr	en	hr	/	**0.5276**	0.3705	0.8181	0.0353
en-hr	hr	en	/	**0.6273**	0.5198	0.9291	0.0812
en-sl	en	sl	**0.6998**	0.4410	0.3812	0.8387	0.0303
en-sl	sl	en	0.6283	**0.6285**	0.4887	0.9291	0.0812

Results show that the size of the dictionary matters. In most cases, we obtain the best results by using the largest dictionary, i.e. the BabelNet dictionary of anchoring points. The exception is the English-Slovene experiment, where in some cases, the smaller BabelNet dictionary slightly outperforms the human-made OES dictionary. As expected, our models produce much higher macro F_1 scores than default value and lower than direct learning. Nevertheless, the results show that the contextual cross-lingual mappings can be successfully trained from automatically generated dictionaries. For the Croatian-Slovene language pair, the difference between the transferred model and directly trained is surprisingly low (2%). How to obtain similarly low scores for other languages remains an open question.

Size of the anchoring dataset

In our last experiment, we systematically test the impact of the dictionary size (and resulting contextual mapping dataset) on cross-lingual transfer performance. For this experiment, we use our largest dictionary, the manually created proprietary Oxford English-Slovene dictionary, as the source of anchoring points. As the parallel corpus, we use OpenSubtitles. We vary the size of the contextual mapping dataset from 1000 to 100 000. For comparison, with the first experiment in this section, we also generated a dataset with the same size as the one made with the low-quality dictionary from Wiktionary. We used the ELMoGAN-10k mapping method, as this was the most successful method on English-Slovene language pair. The results are presented in Figure 2.

Figure 2 & Table 6: Comparison of different sizes of cross-lingual contextual datasets based on different dictionaries used for cross-lingual mapping of contextual ELMo embeddings, evaluated on the NER task. LQsize represents the size of the dataset based on the low quality dictionary (89 800 entries). The mapping method used was ELMoGAN-10k.



Train lang.	Eval. lang.	size	macro F_1
English	Slovenian	1k	0.073
English	Slovenian	2k	0.151
English	Slovenian	5k	0.275
English	Slovenian	10k	0.401
English	Slovenian	20k	0.450
English	Slovenian	50k	0.625
English	Slovenian	100k	**0.646**
English	Slovenian	LQsize	0.592
Slovenian	English	1k	0.106
Slovenian	English	2k	0.197
Slovenian	English	5k	0.243
Slovenian	English	10k	0.377
Slovenian	English	20k	0.497
Slovenian	English	50k	0.564
Slovenian	English	100k	**0.575**
Slovenian	English	LQsize	0.573

Again, the results show that the size of the cross-lingual contextual dataset matters. The macro F_1 score steadily rises, and even 100 000 is not the upper limit, as the results in Table 5 show for the full size of the dataset based on the OES dictionary (294 318 entries).

2.3.2 Dependency parsing

To make the results more general and not specific only to the NER task, we evaluate the proposed contextual cross-lingual mappings on the dependency parsing (DP) task. Similarly to the NER task (Section 2.3.1), we are only interested in using the task for comparison of different mapping methods and not to improve the parsing performance maximally.

The dependency parsing task constructs a dependency tree of a given sentence. In DP, all the words in a sentence are arranged into a hierarchical tree based on their semantic dependencies. Each word has at most one parent node, and only the root word has no parent. A word can have multiple children nodes. In addition to predicting the structure of the tree, the task is also to label the hierarchical dependencies.

As the dependency parsing architecture, we use the SuPar tool by Yu Zhang[9], which is based on the deep biaffine attention (Dozat & Manning, 2017). We modified the SuPar tool to accept ELMo embeddings on the input; specifically, we used the concatenation of the three ELMo layers. The modified

[9] https://github.com/yzhangcs/parser

code is available as one of the associated outputs, listed in Section 6. We trained the parser for 10 epochs, using datasets in nine languages (Croatian, English, Estonian, Finnish, Latvian, Lithuanian, Russian, Slovene, and Swedish). The datasets are obtained from the Universal Dependencies (Nivre et al., 2020). The number of sentences and tokens is shown in Table 7.

We use two evaluation metrics in the dependency parsing task, the mean of unlabeled and labelled attachment scores (UAS and LAS) on the test set. The UAS and LAS are standard accuracy metrics in DP. The UAS score is defined as the proportion of tokens that are assigned the correct syntactic head, while the LAS score is the proportion of tokens that are assigned the correct syntactic head as well as the dependency label (Jurafsky & Martin, 2009).

We evaluated the trained parsers in a cross-lingual manner, using four cross-lingual mapping approaches: ELMoVM, ELMoMU, ELMoGAN-O, and ELMo-GAN-10k, described in Section 2.2.3.

Table 7: Dependency parsing datasets and their properties: the treebank, number of sentences, number of tokens, and information about the size of the split.

Language	Treebank	Tokens	Sentences	Train	Validation	Test
Croatian	SET	199409	9010	6914	960	1136
English	EWT	254855	16622	12543	2002	2077
Estonian	EDT	438171	30972	24633	3125	3214
Finnish	TDT	202697	15135	12216	1364	1555
Latvian	LVTB	220536	13643	10156	1664	1823
Lithuanian	ALKSNIS	70051	3642	2341	617	684
Russian	GSD	98000	5030	3850	579	601
Slovene	SSJ	140670	8000	6478	734	788
Swedish	Talbanken	96858	6026	4303	504	1219

Comparison of novel contextual cross-lingual mapping techniques

Similarly to the NER task, we construct a prediction model for the DP task in the source language for each tested mapping method and each of the tested languages. The prediction model is then evaluated on the target language dataset, using target language ELMo embeddings, mapped to the source language vector space. We report the results in Table 8.

The ELMo-VM mapping method outperforms both ELMoGAN methods on all language pairs in this task. Larger dictionaries, created with triangulation, performed better than smaller direct dictionaries, despite the triangulated dictionaries being of worse quality. Language pairs with similar languages performed better than when the training language was English. The exception is the evaluation on Latvian, where the model trained on English performed better than the model trained on Lithuanian. For evaluation on Lithuanian, both models, trained on English and Latvian, outperform the Lithuanian model. This indicates a poorly trained Lithuanian model, which explains the aforementioned exception in the evaluation of Latvian. The poor Lithuanian model can be partially explained by the small size of the Lithuanian treebank dataset, as seen in Table 7.

The ELMoMU method is stable on the DP task, which is not the case on the NER task. ELMoMU performs on par with ELMo-VM on a few language pairs. Still, its results lie somewhere between ELMo-VM and ELMoGAN on average.

A downside of the ELMo-VM mapping method is that we have to train a model on the downstream task for each pair of mapped languages. This weakness is due to the use of vecmap alignment, which changes embedding vectors of both languages in the process of their mapping. The vecmap method first normalizes all word vectors of a language pair. Then it calculates the mapping matrix, which maps vectors from one language to the other language. Finally, it reweighs both sets of vectors. Because both the target and source language vectors are changed with this method, we have to train a dependency parsing model for each language pair. In contrast to that, for the ELMoGAN method, we trained

Table 8: Comparison of different contextual cross-lingual mapping methods on dependency parsing task. Results are reported as unlabeled attachments score (UAS) and labeled attachment score (LAS). "Direct" stands for direct learning on the target (ie. eval.) language without cross-lingual transfer

Train	Eval.		ELMoVM		ELMoGAN-O		ELMO-GAN-10k		ELMoMU		Direct	
lang.	lang.	Dict.	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
en	hr	direct	**73.96%**	**60.53%**	69.75%	50.20%	65.66%	38.95%	71.01%	54.89%	91.74%	85.84%
en	et	direct	**62.08%**	**40.62%**	52.75%	30.05%	43.48%	22.97%	58.76%	34.07%	89.54%	85.45%
en	fi	direct	**64.40%**	**45.32%**	49.41%	29.35%	42.54%	22.69%	55.03%	37.61%	90.83%	86.86%
en	lv	direct	**77.84%**	**65.97%**	68.43%	46.09%	67.30%	38.38%	76.26%	63.45%	88.85%	82.82%
en	lt	direct	**67.92%**	**39.62%**	56.60%	30.19%	62.26%	24.53%	66.04%	37.74%	55.05%	24.39%
en	ru	direct	**72.00%**	**16.62%**	66.46%	9.23%	61.85%	8.31%	/	/	89.33%	83.54%
en	sl	direct	**79.01%**	**59.84%**	68.38%	48.87%	64.98%	44.86%	77.18%	56.53%	93.70%	91.39%
en	sv	direct	82.08%	72.74%	74.45%	60.39%	75.14%	60.69%	**82.17%**	**72.78%**	89.70%	85.07%
hr	sl	direct	**85.47%**	**72.70%**	54.06%	34.17%	55.34%	32.77%	83.45%	69.08%	93.70%	91.39%
hr	sl	triang	**87.70%**	**76.51%**	73.23%	60.95%	70.86%	54.62%	**87.70%**	76.40%	93.70%	91.39%
et	fi	direct	**79.14%**	**66.09%**	52.97%	34.25%	49.68%	28.37%	76.66%	60.01%	90.83%	86.86%
et	fi	triang	**80.94%**	**67.35%**	54.91%	31.94%	54.40%	26.91%	76.96%	63.37%	90.83%	86.86%
fi	et	direct	**75.81%**	57.32%	54.23%	34.19%	54.64%	32.90%	74.96%	**58.14%**	89.54%	85.45%
fi	et	triang	**79.04%**	**61.86%**	61.19%	39.46%	56.41%	32.58%	76.74%	60.27%	89.54%	85.45%
lv	lt	direct	**72.38%**	**51.43%**	64.76%	45.71%	61.90%	35.24%	67.62%	50.48%	55.05%	24.39%
lv	lt	triang	**75.24%**	50.48%	68.57%	39.05%	69.52%	34.29%	74.29%	**53.33%**	55.05%	24.39%
lt	lv	direct	**63.68%**	**25.88%**	43.46%	11.99%	52.43%	13.54%	61.05%	18.87%	88.85%	82.82%
lt	lv	triang	**61.86%**	**25.94%**	43.13%	9.23%	52.43%	13.68%	57.95%	17.45%	88.85%	82.82%
sl	hr	direct	**77.89%**	**62.58%**	49.36%	29.93%	51.01%	32.03%	72.87%	55.70%	91.74%	85.84%
sl	hr	triang	**81.32%**	**67.51%**	75.02%	56.90%	69.78%	48.94%	78.63%	63.96%	91.74%	85.84%

the English model on English data without any additional mapping and only applied mappings to each language during evaluation. For ELmoVM, we had to train eight different English models on English data, one for each mapped language (the English vectors change with every language in the pair). The computed changes are applied during the evaluation of the models. This considerably slows down the procedure. We tested four alternative approaches: 1) we removed the normalization performed during the evaluation, 2) we removed the normalization during the mapping matrix calculation and evaluation, 3) we added the normalization during the evaluation but did not use it during the calculation of mapping matrix, and 4) we used the normalization both during the evaluation and mapping matrix calculation. None of the approaches was successful, so we omit the results and leave the topic for further investigation.

Using NER and BabelNet as anchor points

To test the impact of different anchor points on the DP task, we use the same settings as in 2.3.1, Results are presented in Table 9.

Table 9: Impact of different dictionaries as sources of anchoring points on the DP task, measured with LAS and UAS scores. We use the ELMoVM mapping method. The best non-reference result for each language pair is in bold. OES stands for human-made Oxford English-Slovene dictionary, BNd for BabelNet dictionary, NERd for dictionary extracted from NER applied to parallel corpora, "direct" for direct learning on the target language without cross-lingual transfer, and "default" for the default scores calculated on the target language.

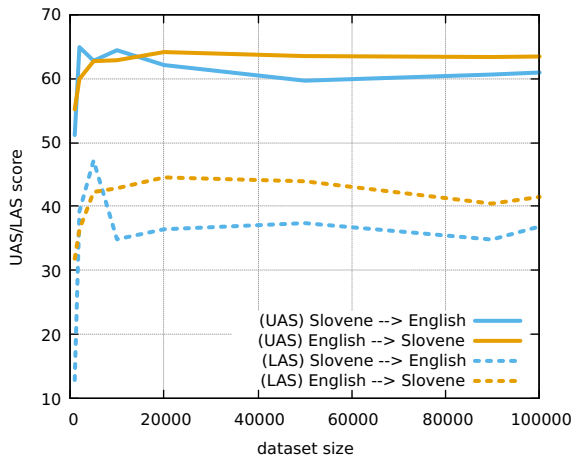
Train	Eval	OSE	OSE	BNd	BNd	NERd	NERd	direct	direct	default	default
lang.	lang.	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
hr	sl	/	/	**66.42%**	**50.34%**	65.97%	50.13%	93.73%	91.37%	5.69%	0.75%
sl	hr	/	/	**84.68%**	70.19%	84.66%	**70.64%**	91.64%	85.73%	4.51%	0.55%
en	hr	/	/	**72.81%**	**47.05%**	70.55%	42.69%	91.64%	85.73%	4.51%	0.55%
hr	en	/	/	42.77%	**13.50%**	**43.62%**	12.70%	92.00%	88.73%	6.52%	0.76%
en	sl	**43.85%**	**17.24%**	41.15%	15.67%	36.98%	12.57%	93.73%	91.37%	5.69%	0.75%
sl	en	41.90%	11.41%	41.68%	10.80%	**42.90%**	**13.76%**	92.00%	88.73%	6.52%	0.76%

Oxford English-Slovene dictionary, BabelNet dictionary, and NER dictionary produce similar results on the DP task for most language pairs. Some notable exceptions are the English-Croatian test, where BabelNet anchors show 2.26% UAS and 4.36% LAS improvement over NER anchors; in the English-Slovene test, the usage of high-quality dictionary outperforms BabelNet anchors, which produce better results than NER anchors, all by margins larger than 2.5% in case of UAS and 1.5% in case of LAS. Correlation between the dictionary's size and the accuracy on the DP task is visible in some tests, but not in all. This means that in some cases, a lower quality dictionary may provide just as much value as a higher quality dictionary. Our models produce much higher scores than the default models. In the case of the Slovene-Croatian pair, the accuracy is close to direct learning. We can conclude that cross-lingual mapping of models is feasible also for DP tasks, especially for close languages.

The size of the anchoring dataset

Using the same settings as in 2.3.1, we analyse the impact of the dictionary size on the quality of cross-lingual DP prediction models. The results are in Figure 3.

Figure 3 & Table 10: Comparison of different sizes of cross-lingual contextual datasets based on different dictionaries used for cross-lingual mapping of contextual ELMo embeddings, evaluated on the DP task. LQsize represents the size of the dataset based on the low quality dictionary (89 800 entries). We used the ELMoGAN-10k mapping method.



Train lang.	Eval. lang.	size	UAS	LAS
English	Slovenian	1k	51.13%	12.94%
English	Slovenian	2k	**64.85%**	39.24%
English	Slovenian	5k	62.72%	**47.17%**
English	Slovenian	10k	64.37%	34.89%
English	Slovenian	20k	62.06%	36.45%
English	Slovenian	50k	59.63%	37.41%
English	Slovenian	100k	60.89%	36.89%
English	Slovenian	LQsize	60.58%	34.84%
Slovenian	English	1k	55.14%	31.87%
Slovenian	English	2k	59.89%	36.41%
Slovenian	English	5k	62.66%	42.27%
Slovenian	English	10k	62.81%	42.85%
Slovenian	English	20k	**64.09%**	**44.53%**
Slovenian	English	50k	63.45%	43.93%
Slovenian	English	100k	63.39%	41.50%
Slovenian	English	LQsize	63.30%	40.45%

Using different sizes of Oxford English-Slovene dictionary shows that larger size does not correlate with better accuracy scores. When trained on English and evaluated on Slovenian, the dictionary of size 2000 performed best, measured by UAS, while the dictionary of size 5000 performed best, measured by LAS. In the opposite language direction, training on Slovene and evaluating on English, the dictionary with size 20 000 performed best. From all the experiments performed with this dictionary (see Table 3), we can see that increasing the dictionary size up to a certain point improves the performance of mapping. Further increasing the size of the dictionary worsens results.

2.4 Discussion on cross-lingual contextual mappings

The contextual cross-lingual isomorphic mapping ELMoVM has proved to be successful and robust. While it sometimes lags behind non-isomorphic mappings, it does not require fine-tuned hyper-parameters. Due to its use of vecmap mapping, its weakness is a requirement to train a new model on a downstream task for each pair of source and target languages. Therefore, this approach is not well-scalable in scenarios where support for the massive cross-lingual transfer of trained models is desired. The isomorphic ELMoMU mapping seems much less stable and successful, but this observation is task and

language-dependent.

Contextual cross-lingual non-isomorphic mapping ELMoGAN is sensitive to the values of training parameters, mostly the learning rate and the number of iterations but may bring superior performance compared to isomorphic mappings. To find a set of well-performing hyperparameters, this method has to be carefully fine-tuned for each task, e.g., the ELMoGAN method outperformed ELMoVM on the NER task but performed worse on the DP task. As this approach is not sufficiently mature, there are still open questions on the methodology for choosing the right number of iterations for each task. The dictionary induction task, we currently use internally, works well for the NER task but seems inappropriate for the dependency parsing task where greater emphasis is on the language's syntactic properties (and not so much on the words as in the NER task).

The work presented in Section 2 is described in full in (Ulčar & Robnik-Šikonja, 2020), attached here as Appendix A.

3 Cross-lingual model transfer for sentiment prediction

As the second aspect of cross-lingual technologies (besides embedding mappings), we analyze an important practical issue of cross-lingual transfer of trained machine learning prediction models between languages. As the example task, we take the practically important task of sentiment prediction. Text annotation is a costly and lengthy operation, with relatively low inter-annotator agreement (Mozetič et al., 2016). Large annotated datasets are costly to produce and, therefore, rare, especially for low-resourced languages. The transfer of already trained models or datasets from other languages would be useful. It would increase the ability to study different text-related phenomena for many more languages than possible today.

Using a large dataset of Twitter sentiment in 13 languages, we compare two practically important approaches to cross-lingual transfer of trained sentiment prediction models. The first is based on mappings of words into a common vector space, as implemented in the LASER library (Artetxe & Schwenk, 2019), and the second approach is the multilingual BERT model (Devlin et al., 2019). We test two variants of the latter approach, the original mBERT, trained simultaneously on 104 languages, and its variants trained on only three languages (Ulčar & Robnik-Šikonja, 2020). The initial version of this work, in the form of a conference paper by Robnik-Šikonja et al. (2020), was reported in D3.2 in the context of user-generated contents and cross-lingual model transfer with LASER library; in this report, we extend this work with the cross-lingual transfer of prediction models with different variants of multilingual BERT. The full description is drafted as a journal publication, available in Appendix B.

Our study's advantage over other studies in the cross-lingual transfer of sentiment prediction models (Wehrmann et al., 2017) are large comparably annotated datasets in 13 different languages, which gives credibility and general validity to our findings. Due to the size of the datasets, we can reliably test both direct transfer between languages (called zero-shot transfer) as well as transfer with sufficient data available for fine-tuning in the target language. The results show a relatively low decrease in predictive performance when transferring trained sentiment prediction models between languages, and superior performance of multilingual BERT models, especially those covering fewer languages. Additionally, we analyse the quality of representations for the Twitter sentiment classification (without cross-lingual transfer). We compare the common vector space for several languages constructed by the LASER library, multilingual BERT, and traditional bag-of-words approach.

This section is divided into five parts. In Section 3.1, we present the background information on the tested cross-lingual technologies, LASER and multilingual BERT. In Section 3.2, we present a large collection of tweets from 13 languages used in the empirical evaluation, the evaluation metrics, and implementation details of our deep neural network prediction models. Section 3.3 contains the experiments on cross-lingual transfer of models and Section 3.4 compares the embedding spaces of LASER,

BERT, and bag-of-words. In Section 3.5, we discuss the results and their implications.

3.1 Cross-lingual transfer technologies

As mentioned in Section 2.1, there are three groups of approaches to find cross-lingual mappings. In Section 2.2, we analyzed the first group of approaches that uses monolingual embeddings with the optional help from a bilingual dictionary to align the embeddings. In this section, we compare the second and third group of cross-lingual transfer technologies. The second group of approaches, presented in Section 3.1.1, uses bilingually aligned (comparable or even parallel) corpora for joint construction of embeddings in all involved languages. The third approach is based on large pretrained multilingual masked language models such as BERT (Devlin et al., 2019), presented in Section 3.1.2.

3.1.1 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 the 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[10] (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 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 many languages; currently, the encoder supports 93 different languages, including all EMBEDDIA languages. The resulting joint embedding can be transformed back into a sentence using a 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.

3.1.2 BERT contextual model

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) generalises the idea of language models to masked language models—inspired by cloze (i.e. gap filling) tests—which test the understanding of a text by removing a certain portion of words that the participant is asked to fill in. The masked language model randomly masks some of the tokens from the input. The task of the language model is to predict the missing token based on its neighbourhood. BERT uses transformer architecture of neural networks (Vaswani et al., 2017), which uses both left and right context in predicting the masked word 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 result of pre-trained tokenization is that some common words are kept as single tokens. In contrast, others are split into subwords (e.g., common stems, prefixes, suffixes—if needed down to single letter tokens). The original BERT project offers pre-trained English, Chinese, and multilingual models; the latter, called mBERT, is trained on 104 languages simultaneously. 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.

Recently, a new type of multilingual BERT models emerged that reduce the number of languages in multilingual models. CroSloEngual BERT (Ulčar & Robnik-Šikonja, 2020), built in T1.2, uses Croatian,

[10] https://github.com/facebookresearch/LASER

Slovene (two similar less-resourced languages from the same language family), and English. The main reasons for this choice are to represent each language better and keep sensible sub-word vocabulary, as shown by Virtanen et al. (2019). CroSloEngual BERT was trained using Wikipedia for English text, the Gigafida corpus for Slovene text, and a combination of hrWaC (Ljubešić & Erjavec, 2011), articles from the Styria media group, and Riznica corpora (Ćavar & Rončević, 2012) for Croatian text. This model is built with the cross-lingual transfer of prediction models in mind. It uses the same architecture and is fine-tuned in the same way as mBERT. By including English, we expect that the CroSloEngual model will enable a better transfer of existing prediction models from English to Croatian and Slovene.

3.2 Datasets and experimental settings

In this section, we present the evaluation metrics, experimental data, and implementation details of the used neural prediction models.

3.2.1 Evaluation metrics

Following Mozetič et al. (2016) we report $\overline{F_1}$ score which takes positive and negative sentiment into account, and classification accuracy CA. $F_1(c)$ score for class value c is the harmonic mean of precision p and recall r for the given class c, where the precision is defined as the proportion of correctly classified instances from the instances predicted to be from the class c, and the recall is the proportion of correctly classified instances actually from the class c.

$$F_1(c) = \frac{2p_c r_c}{p_c + r_c}.$$

The F_1 score returns values from $[0, 1]$ interval, where 1 means perfect classification and 0 completely wrong predictions. We use F_1 score averaged over positive $(+)$ and negative $(-)$ sentiment class:

$$\overline{F_1} = \frac{F_1(+) + F_1(-)}{2}.$$

As the sentiment labels are ordered, the neutral sentiment label is implicitly taken into account in $\overline{F_1}$.

The classification accuracy CA is defined as the ratio of correctly predicted tweets N_c to all the tweets N:

$$CA = \frac{N_c}{N}$$

3.2.2 Datasets

We use a corpus of Twitter sentiment datasets (Mozetič et al., 2016), consisting of 15 languages, with over 1.6 million annotated tweets. The languages covered are Albanian, Bosnian, Bulgarian, Croatian, English, German, Hungarian, Polish, Portuguese, Russian, Serbian, Slovak, Slovene, Spanish, and Swedish. The authors studied the annotators' agreement on the labelled tweets. They discovered that the SVM classifier achieves a significantly lower score for some languages (English, Russian, Slovak) than the annotators. This hints that there might be room for improvement for these languages using a better classification model or larger training set.

We cleaned the above datasets by removing the duplicated tweets, weblinks, and hashtags. Due to the low quality of sentiment annotations indicated by low self-agreement and low inter-annotator agreement, we removed Albanian and Spanish datasets. For these two languages, the self-agreement expressed with $\overline{F_1}$ score (i.e. $F_1(c)$ is the fraction of equally labelled tweets out of all the tweets with a given label c) is 0.60 and 0.49, respectively; the inter-annotator agreement is 0.41 and 0.42. The characteristics of the remaining 13 datasets are presented in Table 11.

	Number of tweets				Agreement	
Language	Negative	Neutral	Positive	All	Self-	Inter-
Bosnian	12,868	11,526	13,711	38,105	0.81	0.51
Bulgarian	15,140	31,214	20,815	67,169	0.77	0.50
Croatian	21,068	19,039	43,894	84,001	0.81	0.51
English	26,674	46,972	29,388	103,034	0.79	0.67
German	20,617	60,061	28,452	109,130	0.73	0.42
Hungarian	10,770	22,359	35,376	68,505	0.76	-
Polish	67,083	60,486	96,005	223,574	0.84	0.67
Portuguese	58,592	53,820	44,981	157,393	0.74	-
Russian	34,252	44,044	29,477	107,773	0.82	-
Serbian	24,860	30,700	16,161	71,721	0.81	0.51
Slovak	18,716	14,917	36,792	70,425	0.77	-
Slovene	38,975	60,679	34,281	133,935	0.73	0.54
Swedish	25,319	17,857	15,371	58,547	0.76	-

Table 11: The left-hand side reports the number of tweets from each of the category and the overall number of instances for individual languages. The right-hand side contains self-agreement of annotators, and inter-annotator agreement for languages where more than one annotator was involved.

3.2.3 Implementation details

In our experiments, we use three different types of prediction models, BiLSTM neural networks using joint vector space embeddings constructed with the LASER library, and two variants of multilingual BERT, mBERT and CroSloEngual BERT.

The cross-lingual embeddings from the LASER library are pretrained on 93 languages, using BiLSTM networks. They are stored as 1024 dimensional embedding vectors. Our classification models contain the embedding layer, followed by the multilayer perceptron hidden layer of size 8, and an output layer with three neurons (corresponding to three output classes, negative, neutral, and positive sentiment) using the softmax. We use the ReLU activation function and Adam optimizer. The fine-tuning uses a batch size of 32 and 10 epochs.

The multilingual mBERT model (Devlin et al., 2019) is case sensitive (i.e. bert_multi_cased), pretrained on 104 languages, has 12 transformer layers, and 110 million parameters. Rather than training an individual classifier for every classification task from scratch, which would be resource and time expensive, the pre-trained BERT language model is usually used and fine-tuned on a specific task. This approach is common in modern NLP because large pretrained language models extract highly-relevant textual features without task-specific development and training. Frequently, this approach also requires less task-specific data. During pre-training, the BERT model learns relations between sentences (entailment) and between tokens within a sentence. This knowledge is used during training on a specific down-stream task (Devlin et al., 2019). The use of BERT for a token classification task requires adding connections between its last hidden layer and new neurons corresponding to the number of classes in the intended task. To classify a sequence, we use a special [CLS] token representing the final hidden state of the input sequence (i.e. the sentence). The predicted class label of the [CLS] token corresponds to the class label of the entire sequence. The fine-tuning process is applied to the whole network. All parameters of BERT and new class-specific weights are fine-tuned jointly to maximize the log-probability of the correct labels.

3.3 Experiments in cross-lingual transfer

Our experimental evaluation focuses on text representations using embeddings into a common vector space with the LASER library and two variants of multilingual BERT, mBERT and CroSloEngual BERT, described in Section 3.1. We report transfer of models between languages from the same language

family (Slavic and Germanic), as here successful transfer is most likely. We did not test all possible combinations of languages, but we give a representative sample. To thoroughly test CroSloEngual BERT, we also test Croatian, Slovene, and English (English not similar to the other two). We report the results in Table 12.

		LASER		mBERT		cseBERT		Both target	
Source	Target	$\overline{F_1}$	CA	$\overline{F_1}$	CA	$\overline{F_1}$	CA	$\overline{F_1}$	CA
German	English	0.55	0.59	**0.63**	**0.64**	0.42	0.42	0.62	0.65
English	German	0.55	0.60	**0.66**	**0.70**	0.50	0.58	0.53	0.65
Polish	Russian	**0.64**	**0.59**	0.57	0.57	0.50	0.40	0.70	0.70
Polish	Slovak	**0.63**	0.59	0.58	0.59	**0.63**	**0.65**	0.72	0.72
German	Swedish	0.58	0.57	**0.59**	**0.59**	0.58	0.56	0.67	0.65
German Swedish	English	**0.58**	**0.60**	0.55	0.56	0.41	0.42	0.62	0.65
Slovene Serbian	Russian	0.53	0.55	0.57	**0.57**	**0.58**	0.48	0.70	0.70
Slovene Serbian	Slovak	**0.59**	0.52	0.57	0.59	0.48	**0.60**	0.72	0.72
Serbian	Slovene	0.54	**0.57**	0.54	0.54	**0.56**	0.55	0.60	0.60
Serbian	Croatian	**0.67**	0.64	0.65	0.62	0.65	**0.70**	0.73	0.68
Serbian	Bosnian	**0.65**	0.61	0.61	0.60	0.59	**0.62**	0.67	0.64
Polish	Slovene	0.51	0.48	**0.55**	**0.54**	0.50	0.53	0.60	0.60
Slovak	Slovene	0.52	0.51	0.54	0.54	**0.58**	**0.58**	0.60	0.60
Croatian	Serbian	**0.54**	**0.52**	0.52	0.51	0.52	0.49	0.48	0.54
Croatian	Bosnian	0.66	0.61	0.57	0.56	**0.67**	**0.62**	0.67	0.64
Slovene	Serbian	**0.52**	**0.55**	0.46	0.49	0.47	0.50	0.48	0.54
Slovene	Bosnian	**0.66**	0.61	0.58	0.56	**0.66**	**0.62**	0.67	0.64
Croatian	Slovene	0.53	0.53	0.53	0.54	**0.61**	**0.60**	0.60	0.60
Slovene	Croatian	0.70	0.65	0.64	0.63	**0.73**	**0.69**	0.73	0.68
English	Slovene	0.54	**0.57**	0.50	0.53	**0.59**	**0.57**	0.60	0.60
Average performance gap		**0.05**	**0.07**	0.06	**0.07**	0.08	0.08		
Average performance gap on CSE		0.05	0.04	0.09	0.06	**0.00**	**0.01**		

Table 12: The transfer of trained models between languages from the same language family using LASER common vector space, and two variants of BERT: original multilingual BERT (mBERT) and CroSloEngual BERT (cseBERT). As a reference, we include the comparison with both training and testing set from the target language (the right-most column). At the bottom, we report the average performance gap across all languages compared to the reference scores and the average gap for only Croatian, Slovene, and English. The best cross-lingual transfer score for each of the languages is in bold.

In each experiment, we use the entire dataset of the source language as the training set and the whole dataset of the target language as the testing set, i.e. we do a zero-shot transfer. We compare the results with the training and testing set from the target language, where 70% of the dataset is used for training and 30% for testing. The latter results can be taken as an upper bound of what the transfer models could achieve in ideal conditions. The results from Table 12 show that there is a performance gap between the transfer learning models and native models. On average, the gap in $\overline{F_1}$ is 5% for LASER approach and 6% for multilingual BERT; for the classification accuracy, the average gap is 7% for both LASER and mBERT. However, there are significant differences between languages. We advise testing both variants for a specific language, as they are highly competitive. The CroSloEngual BERT is slightly less successful measured with an average performance gap over all languages: the gap is 8% in both $\overline{F_1}$ and CA. However, if we take only the three languages used in the training of CroSloEngual BERT (the last three language pairs in Table 12), the conclusions are completely different. The average performance gap is 0% in $\overline{F_1}$ and 1% in classification accuracy, meaning that we get an almost perfect cross-lingual transfer for these languages on the Twitter sentiment prediction task.

3.4 Comparing representations

We also compare different types of representations commonly used in text classification: embeddings into a common vector space obtained with LASER library, the multilingual BERT model, and bag-of-ngrams representation with SVM classifier. Note that there is no transfer between different languages in this experiment but only a test of the quality of the representation, i.e. embeddings. The training set in each experiment consists of randomly chosen 70% of the dataset for each language, while the remaining 30% of instances are used as the testing set. The SVM model does not use neural embeddings, but the Delta TF-IDF weighted bag-of-ngrams representation with substantial preprocessing of tweets (Mozetič et al., 2016). The results were obtained with 10-fold blocked cross-validation. The datasets for Bosnian, Croatian, and Serbian languages were merged in (Mozetič et al., 2016) due to the similarity of these languages. We report the performance on the merged dataset for the SVM classifier. Results are presented in Table 13.

	LASER		mBERT		SVM	
Language	$\overline{F_1}$	CA	$\overline{F_1}$	CA	$\overline{F_1}$	CA
Bosnian	**0.67**	0.64	0.65	**0.66**	0.61	0.56
Bulgarian	0.50	0.59	**0.58**	**0.60**	0.52	0.54
Croatian	**0.73**	**0.68**	0.64	**0.68**	0.61	0.56
English	0.62	0.65	**0.72**	**0.71**	0.63	0.64
German	0.53	0.65	**0.66**	**0.66**	0.54	0.61
Hungarian	0.60	0.67	**0.65**	**0.69**	0.64	0.67
Polish	**0.70**	0.66	**0.70**	**0.73**	0.68	0.63
Portugal	0.52	0.51	**0.66**	**0.67**	0.55	0.51
Russian	0.70	0.70	**0.74**	**0.75**	0.61	0.60
Serbian	0.48	0.54	0.56	0.54	**0.61**	**0.56**
Slovak	**0.72**	0.72	0.70	**0.75**	0.68	0.68
Slovene	0.60	0.60	**0.66**	**0.64**	0.55	0.54
Swedish	**0.67**	0.65	0.64	**0.66**	0.66	0.62
Average	0.62	0.64	**0.66**	**0.67**	0.61	0.59

Table 13: Comparison of different representations: supervised mapping into a common vector space with the LASER library, multilingual BERT, and bag-of-ngrams with the SVM classifier. The best score for each language and metric is in bold.

The SVM baseline using bag-of-ngrams representation achieves on average lower predictive performance than the two neural embedding approaches. We believe that the main reason for this is the knowledge about the language structure contained in large precomputed embeddings used by the neural approaches. Together with the fact that standard feature-based machine learning approaches require much more preprocessing effort, it seems that there are no good reasons why to bother with this approach in text classification. The multilingual BERT is the best of the three tested methods, achieving the best average $\overline{F_1}$ and CA scores and the best result in most languages (in bold). The downside is that the fine-tuning and execution of mBERT requires more computational time than precomputed fixed embeddings. Nevertheless, with progress in optimization techniques for neural network learning and the advent of computationally more efficient BERT variants, e.g., (You et al., 2020), this obstacle might disappear in the future.

3.5 Discussion on cross-lingual sentiment predictors transfer

We studied two approaches to the cross-lingual transfer of Twitter sentiment prediction models. LASER approach is based on mappings of words into the common vector space, and multilingual BERT models are trained on a multitude of languages: mBERT on 104 languages, and CroSloEngual BERT (trained in T1.2) on three languages. Our empirical evaluation is based on relatively large datasets of labelled tweets from 13 European languages.

Our results show a significant transfer potential using the models trained on similar languages; compared to training and testing on the same language, we get on average 5% lower $\overline{F_1}$ score with LASER and 6% with mBERT. Using a specialized trilingual variant of BERT, we get an even better cross-lingual transfer. Using CroSloEngual BERT for cross-lingual model transfer in the three involved languages, we get on average the same $\overline{F_1}$ score and 1% lower classification accuracy compared to training and testing in the same language with the LASER library. This result shows that the specialized BERT models enable almost zero-loss cross-lingual transfer to less-resourced languages, at least for certain tasks.

The comparison of the quality of three text representations, cross-lingual joint embedding space of LASER library, multilingual BERT embeddings, and classical bag-of-ngram representation coupled with SVM classifier, shows that the multilingual BERT is the most successful of the three, followed by the common vector space of LASER, while SVM with bag-of-ngrams is rarely competitive.

The work presented in Section 3 is described in full in (Robnik-Šikonja et al., 2020), attached here as Appendix B.

4 Cross-lingual transfer for prediction of idioms

This section demonstrates the practical use of modern contextual and cross-lingual embeddings, developed in T1.1 and T1.2, in a monolingual and multilingual setting. As a testbed, we use a practically important linguistic problem, namely the detection of idiomatic expressions. In contrast to experiments in Section 2 and Section 3, where we used well-know tasks and datasets, in this section, we show that contextual embeddings can be applied to entirely new task, where previous technologies were not successful. We show that deep neural networks using either ELMo or BERT embeddings (produced in T1.2) perform better than existing approaches and can detect idiomatic word use even for idioms that were not present in the training set. We develop a novel dataset of idiomatic expressions, demonstrate the cross-lingual transfer of developed models, and show that contextual word embeddings can generalize to other languages. This work is relevant for T1.2, as it shows the difference in using fixed ELMo and BERT vectors compared to fine-tuning of the entire models. The successful detection of idiomatic expressions has the potential to be applied in T5.3 and improve approaches for creative language use, in particular for metaphors.

Multiword expressions (MWEs) are made up of at least two words that can be syntactically and/or semantically idiosyncratic in nature and can cover fixed and nonfixed expressions. They often act as a single unit in linguistic analysis. MWEs are not homogeneous and are commonly divided in the literature into idiomatic expressions, light-verb constructions, verb-particle constructions, complex function words, multiword named entities, and multiword terms (Constant et al., 2017). According to Jackendoff (1997), the number of MWEs in a speaker's lexicon may be "of the same order of magnitude as the number of single words of the vocabulary", and according to Sag et al. (2002) "it seems likely that this is an underestimate". This emphasizes the importance of good coverage and accuracy of MWE detection and identification tools for NLP applications and lexical resources.

In this section, we are interested in the detection and identification of idiomatic expressions (IEs), also called idioms, that are composed of a group of lexemes whose meaning is established by convention and cannot be deduced from individual lexemes composing the expression (e.g., it's a piece of cake).

Due to the lack of satisfactory tools, linguists often create lexicons of idioms manually or use tools that take into account only co-occurrence features since these are easier to implement and are relatively language independent. This type of workflow introduces several problems. First, manually created large lexicons of idioms are scarce because of the time-consuming human labor required, particularly for less-resourced languages. Second, frequency lists of idioms created without robust, generalized identification tools are unreliable – mostly due to their discontinuity and syntactic variability. Finally, discovering or detecting new IEs is often based on linguists' personal knowledge or frequent collocations. This may completely omit many idioms.

IEs such as "break the ice" and "under the weather" commonly occur in texts. They can be hard to understand for computer models as their meaning differs from the meaning of individual words. To address this, several automatic machine learning-based approaches for the detection of idiomatic language emerged. However, current approaches suffer from several issues and limitations related to methodological shortcomings and a lack of datasets. The first issue that affects current approaches is the lack of large datasets with annotated IEs. Because of a large number of different IEs, a dataset that would contain a sufficient number of examples for every IE needed to train a classification model currently does not exist. Additionally, most existing datasets only address English, which makes developing approaches for other languages difficult. Existing works use small datasets, such as the data from SemEval 2013, task 5B (Korkontzelos et al., 2013), PARSEME Shared Task on Automatic Verbal Multi-Word Expression (MWE) Identification (Savary et al., 2017), or the VNC-tokens dataset (Cook et al., 2008). These datasets only cover a limited number of IEs and contain at most a few annotated sentences for each expression, making it hard to train successful machine-learning models for IE recognition.

In our work, we use currently the most successful approaches to contextual word embeddings, ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). We examine whether contextual word embeddings can be used as a solution to the idiom identification problem. Past work shows that contextual word embeddings can detect different meanings of polysemous words and improve the performance on a variety of NLP tasks (Devlin et al., 2019). However, current approaches have not used contextual word embeddings for differentiating between idiomatic and literal language use. In the proposed approach, called MICE (Mining Idioms with Contextual Embeddings), we use ELMo and BERT embeddings as an input to a neural network and show that using them as the first layer of neural networks improves results compared to existing approaches. We evaluate our approach on a new dataset of Slovene IEs and the existing dataset from the PARSEME Shared Task on Automatic Verbal MWE Identification. We analyze different properties of the proposed models, such as different variants of BERT models and cross-lingual transfer of trained models.

We show that contextual embeddings contain a large amount of lexical and semantic information that can be used to detect IEs. Our MICE approach outperforms existing approaches that do not use pretrained contextual word embeddings in the detection of IE present in the training data, as well as identification of IE missing in the training set. The latter is a major problem of existing approaches. Finally, we show that multilingual contextual word embeddings can detect IEs in multiple languages even when trained on a monolingual dataset.

We present our MICE methodology in Section 4.1. Section 4.2 describes the datasets used for the evaluation of our approach, which we describe in Section 4.3. Section 4.4 discusses results and their implications.

4.1 Novel MICE architecture for idiom detection

The proposed approach is based on contextual word embeddings, which were designed to deal with the fact that a word can have multiple meanings. Instead of assigning the same vector to every occurrence of a word, contextual embeddings assign a different vector to each word occurrence based on its context. As the contexts of words' literal use and idiomatic occurrences of the same word are likely to differ, these embeddings shall be well-suited for detecting IEs. We used two state-of-the-art embedding approaches: ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). For ELMo, we used the pretrained Slovene model described by Ulčar & Robnik-Šikonja (2020a). The model was trained on the Gigafida corpus (Krek et al., 2016) of Slovene texts. For BERT embeddings, we use two models, described in Section 3.1.2: the original multilingual mBERT model presented by Devlin et al. (2019), which was trained on Wikipedia text from 104 languages, including Slovene and Croatian, and the trilingual CroSloEngual BERT presented by Ulčar & Robnik-Šikonja (2020). This BERT is better suited for classification tasks in Slovene and Croatian as mBERT as its training incorporated larger amounts of training data and a larger vocabulary for each of the involved languages. The authors also report an improved cross-lingual transfer of trained models between the three languages.

We use the embeddings (ELMo or BERT) as the first layer of a neural network. This layer is followed by a bidirectional gated recurrent unit (GRU) with 100 cells. GRUs are similar to standard recurrent units but use an additional update and reset gate to help deal with the vanishing gradient problem. The update gate is defined as

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1} + b_z), \tag{2}$$

where $W^(z)$ and $U^(z)$ are trainable weights, x_t is the input vector and b_z is the trainable bias. h_{t-1} represents the memory of past inputs computed by the network. The reset gate uses the same equation, with different weights and biases:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1} + b_r). \tag{3}$$

For each input, the GRU computes the output as:

$$h_t = z_t \odot h_{t-1} + (1 - z) \odot \tanh(W^{(h)}x_t + U^{(h)}(r_t \odot h_{t-1}) + b_h), \tag{4}$$

where $\odot$ is the Hadamard product, and $W^{(h)}$, $U^{(h)}$, and b_h are trainable weights and biases.

For both embeddings used, ELMo and BERT, the GRU layer is followed by a softmax layer to obtain the final predictions. A dropout of 50% is applied at the softmax layer. This approach follows the work on MWE detection presented by Klyueva et al. (2017) but with the difference that we use contextual embeddings. We deliberately use a simple network architecture to show that the embeddings, by themselves, capture enough semantic information to properly recognize IEs.

We use the described architecture on two types of classification tasks: a token-level classification, where we predict whether an individual token has an idiomatic or literal meaning, and a sentence-level classification, where the network makes a single prediction for the entire sentence, predicting whether the sentence contains an expression with an idiomatic meaning. The details of the tasks are presented in Section 4.3.

We fine-tuned the hyperparameters using a development set consisting of 7% of sentences randomly selected from our dataset, described in Section 4.2.1. We trained the network for 10 epochs using RMSProp as the optimizer with the learning rate of 0.001, $\rho = 0.9$, and $\epsilon = 10^{-7}$. We used binary cross-entropy as the loss function.

4.2 Datasets of idiomatic expressions

Our approach supports two types of tasks, monolingual and multilingual. The monolingual approach requires a reasonably large dataset with a sufficient number of idioms. The multilingual approach exploits the existing monolingual dataset to transfer the trained model to languages with fewer resources, i.e. with non-existent or smaller datasets.

In Section 4.2.1, we describe our monolingual Slovene dataset. In Section 4.2.2 we describe the well-known PARSEME datasets (Savary et al., 2017) for detection of multi-word expressions, including idioms, in many languages.

4.2.1 Novel monolingual dataset

We evaluate our approach on a new dataset of Slovene IEs, called SloIE, which we make publicly available for further research[11]. The dataset consists of 29,400 sentences extracted from the Gigafida corpus (Krek et al., 2016) and contains 75 different IEs. The 75 IEs were selected from the Slovene Lexical Database (Gantar & Krek, 2011). They had to meet the condition that they appear in corpus sentences in both idiomatic and literal senses, such as, e.g., break the ice or step on someone's toes. Manual selection of idiomatic examples showed that most IEs in the Slovene Lexical Database (2,041

[11] http://hdl.handle.net/11356/1335

in total) appear more frequently or even exclusively in their idiomatic meaning, either because literal use is not possible (e.g., get under someone's skin), or it's very rare, although possible in terms of syntax and semantics (e.g., to do something behind someone's back). Although this finding is interesting from a (socio)linguistic point of view, in designing the dataset for our purposes, we assumed that the literal and idiomatic interpretation of an expression could be disambiguated by its context.

Two annotators, students of linguistics, marked the complete set of 29,400 sentences. They had four possible choices: YES (the expression in a particular sentence is used in the idiomatic sense, NO (the expression is used in the literal sense), DON'T KNOW (not sure whether the expression is used in a literal or idiomatic sense) and VAGUE, (literal or idiomatic use cannot be inferred from the sentence). Student annotators were previously briefed with short instructions and provided with a sample of good examples. For the training of classification models, we selected only sentences where both annotators agreed on the annotation. The inter-annotator agreement across the entire dataset was 0.952.

Due to the nature of IEs, our dataset is imbalanced. A few expressions occur proportionally in both literal and idiomatic use, while most expressions occurring predominately idiomatically. The dataset contains fewer than 100 occurrences for most expressions. Table 14 shows an overview of the data present in our dataset. The distribution of literal and idiomatic uses of each expression is shown in Figure 4.

Table 14: An overview of the data present in the SloIE dataset.

Sentences	29,400
Tokens	695,636
Idiomatic sentences	24,349
Literal sentences	5,051
Idiomatic tokens	67,088
Literal tokens	626,707
Different IEs	75

SloIE is much larger than other existing datasets of IEs in terms of the number of sentences, e.g., VNC-tokens contains 2,984 instances of 53 IEs. Such a dataset would require significant effort to create for other languages. Nevertheless, as our experiments in (Škvorc et al., 2020) show (see Appendix C), even smaller datasets could produce useful results. Further, there is a significant potential for cross-lingual transfer of trained models, especially between similar languages.

4.2.2 PARSEME datasets

The dataset for the Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions (MWEs) consists of 280,838 annotated sentences split across 20 languages. The corpus contains annotations for various types of verbal MWEs, such as verb-particle constructions, inherently reflexive verbs, and verbal idioms. As our work focuses on detecting IEs, we only predict tags of verbal idioms and ignore all other tags. A summary of the number of sentences for each language used in our work is presented in Table 15. We do not use the Arabic dataset as it was not made available under an open licence.

IEs in the PARSEME datasets only occur in a small number of sentences. Additionally, most IEs occur only once in the corpus, which makes training a classifier difficult. For that reason, we used the PARSEME dataset to evaluate our cross-lingual model. The prediction model used the pretrained mBERT embeddings (Devlin et al., 2019) and was further trained on our Slovene SloIE dataset. It was then tested on each of the PARSEME datasets in different languages. The details are reported in Section 4.3.3.

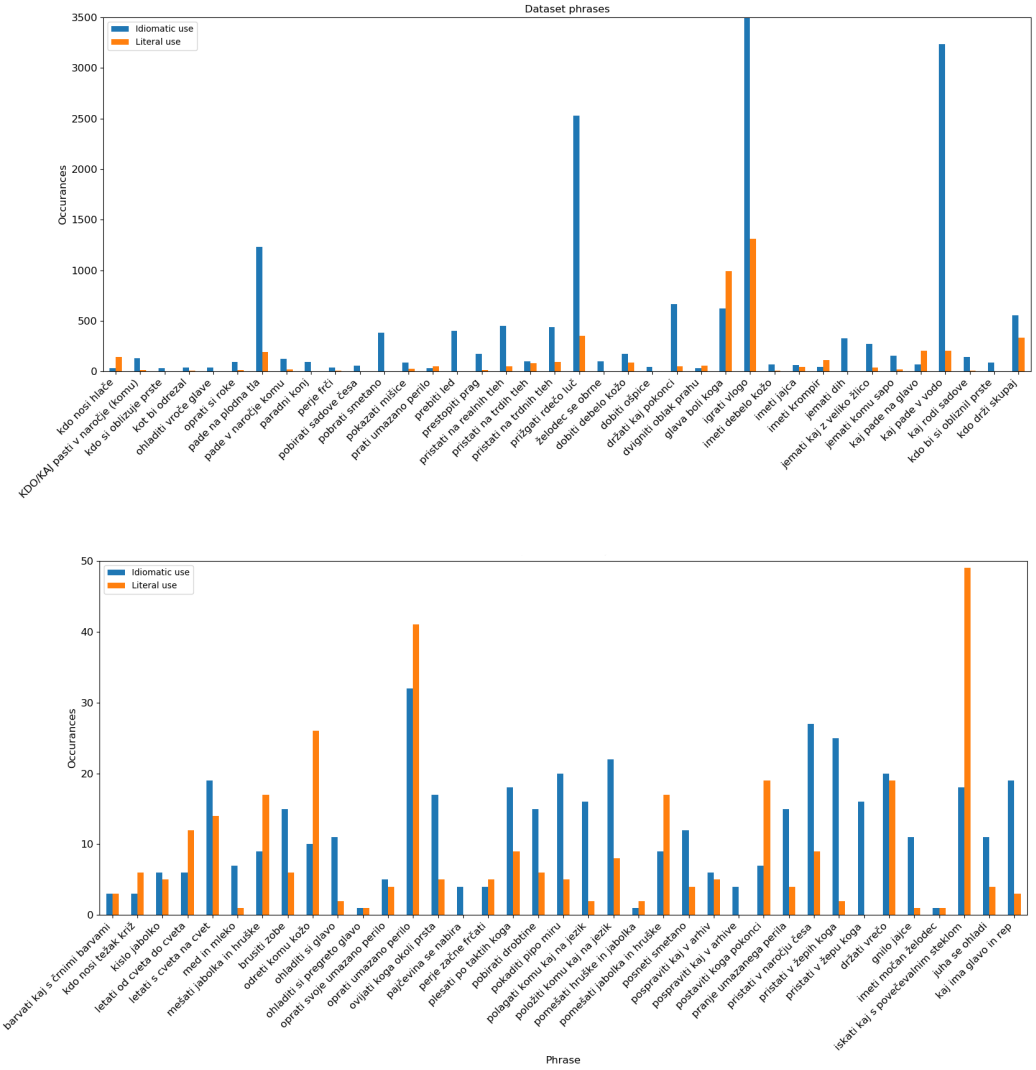


Figure 4: The number of literal and idiomatic uses for IEs present in the SloIE dataset. The top figure shows IEs that occur more than 35 times with an idiomatic meaning. The bottom figure shows IEs that occur less than 35 times with an idiomatic meaning.

4.3 Evaluation

We evaluate our MICE approach in three settings, described below.

1. *Classification of IEs that were present in the training set.* In Section 4.3.1 we evaluate whether MICE is capable of detecting IEs that were present in the training set. We split this task into two subtasks: i) sentence-level classification, where the network makes a single prediction for the entire sentence, predicting whether that sentence contains an expression with the idiomatic meaning, and ii) token-level classification, where we predict whether each token has a literal or idiomatic meaning. The sentence-level classification task is easier, but the token-level task can be more useful, as it can be used to detect which tokens have the idiomatic meaning.
2. *Classification of IEs that were not present in the training set.* Due to many idioms, it is difficult and

Table 15: An overview of the data present in the PARSEME datasets. Of the 20 languages in the PARSEME corpus, we use 18. We omit Arabic because it is not available as an open language, and Farsi, which does not contain IEs. On average, each language contains 586 IEs.

Language	Sentences	Tokens	IEs
BG	6,913	157,647	417
DE	6,261	120,840	1,005
EL	5,244	142,322	515
EN	7,436	124,203	59
ES	2,502	102,090	196
FA	2,736	46,530	0
FR	17,880	450,221	1,786
HE	4,673	99,790	86
HU	3,569	87,777	92
HR	3003	69915	131
IT	15,728	387,325	913
LT	12,153	209,636	229
MT	5,965	141,096	261
PL	11,578	191,239	317
PT	19,640	359,345	820
RO	45,469	778,674	524
SL	8,881	183,285	283
SV	200	3,376	9
TR	16,715	334,880	2,911
Total	19,6546	3,990,191	10,554

expensive to annotate a dataset that would cover every idiom. Thus, it would be desirable that the prediction model is capable of detecting expressions that are not present in the training set. We test this setting in Section 4.3.2. As with the first task, we use sentence-level and token-level classifiers. This task is more difficult than detecting IEs present in the dataset. It can only be solved successfully if the contextual word embeddings contain information about idiomatic word use (e.g., as directions in the vector space).

3. *Cross-lingual transfer on PARSEME dataset.* In Section 4.3.3, we evaluate whether our approach can be used to detect expressions in different languages when trained with multilingual word embedding models. For testing this hypothesis, we use 18 languages from the multilingual PARSEME dataset.

We compare the proposed MICE approach to different existing approaches. As a baseline, we use the SVM classifier with the tf-idf weighted vector of a sentence as an input. We compare our approach to MUMULS (Klyueva et al., 2017), which uses a similar neural network architecture to our approach but does not use pretrained contextual word embeddings. Unlike our approach, MUMULS uses part-of-speech tags and word lemmas as additional inputs.

For all tests, we report the classification accuracy (CA) and F_1 score. As many of the tasks are highly imbalanced, CA is not a good measure, and we mostly use the obtained F_1 scores in interpretations of the results.

4.3.1 IEs from the training set

For the first experiment, detection of IEs present in the training set, we randomly split the SloIE dataset into training, testing, and development sets with the ratio of 63:30:7 (18,522, 8,820, and 2,058 sentences). The network was trained for 10 epochs using RMSProp as the optimizer with the learning rate of 0.001, $\rho = 0.9$, and $\epsilon = 10^{-7}$. The binary cross-entropy was used as the loss function. We report two

sets of results: recognition of individual tokens in a sentence as idiomatic or non-idiomatic (i.e. token-level classification) and detecting the whole sentence as either containing or not containing idioms (i.e. sentence-level classification).

The results for token-level classification are presented in Table 16. To provide a sensible context for token-based classification, the SVM classifier's input consists of the target token and three words before and three words after the target word. The SVM classifier obtains better F_1 score than MUMULS but lower score compared to MICE variants. The dataset is highly imbalanced, with 96,7% of all tokens being non-idiomatic. Lacking discriminating information, MUMULS predicts almost every token as non-idiomatic, which results in high classification accuracy but very low F_1 score. Due to the dataset's imbalanced nature, the F_1 score is more reflective of relevant real-world performance. Here, the MICE variants are in a class of their own.

Table 16: Comparison of results when classifying tokens with the same IEs present in the training and testing set. Each token was classified as either belonging to IE with the literal meaning, belonging to IE with the idiomatic meaning, or not belonging to IE.

Method	CA	F_1
Default classifier	0.903	0.176
SVM baseline	0.8756	0.3962
MUMULS	**0.975**	0.0659
MICE with Slovene ELMo	0.889	**0.9219**
MICE with mBERT	0.814	0.4556
MICE with CroSloEngual BERT	0.972	0.837

Of the three MICE approaches, the Slovene ELMo model obtains the highest F_1 score. The MICE variants with BERT embeddings obtain lower classification accuracies and F_1 scores. This is likely due to different tokenization approaches used by the embeddings. We used ELMo embeddings by first performing word-level tokenization while BERT splits words into sub-word units. Token-level classification with BERT must classify sub-word units instead of classifying entire words, as is the case with ELMo. Additionally, our ELMo embeddings were pretrained on a large amount of only Slovene texts, while the mBERT model was trained on 104 different languages. Only a small amount of Slovene texts was included in its training, and it has a small proportion of Slovene words in the vocabulary. The CroSloEngual embeddings were trained on a larger amount of Slovene text and therefore achieved better results.

In the evaluation on the sentence-level, instead of classifying each token, we classified each sentence whether it contains an IE or not. This lowers the importance of different tokenization strategies between ELMo and BERT. However, the sentence-level evaluation does not show whether the approaches can detect specific words in a sentence as idioms. The results of this evaluation are presented in Table 17.

Table 17: Comparison of results when classifying sentences from the SloIE dataset and the same IEs are present in the training and testing sets. Each sentence was classified as either containing an expression with the literal meaning or containing an expression with the idiomatic meaning.

Method	CA	F_1
Default classifier	0.828	0.906
SVM baseline	0.900	0.942
MUMULS	0.915	0.948
MICE with Slovene ELMo	**0.951**	**0.980**
MICE with mBERT	0.897	0.908
MICE with CroSloEngual BERT	0.921	0.954

The sentence-level classification task is easier than the token-level task, which leads to an improved performance for all models. The SVM baseline outperforms the mBERT model. MUMULS also achieves

better results, outperforming the SVM baseline and the mBERT approach. MICE with CroSloEngual BERT is closer to ELMo in this task, though the latter still achieves the best scores. MICE with mBERT likely achieves lower scores because this model was not pretrained on a large enough amount of Slovene texts.

4.3.2 IEs outside the training set

In the previous experiment with the same IEs present in both the training and testing set, we obtained good results (especially with the ELMo contextual embeddings). However, many languages lack large annotated datasets and even when they do exist, they are unlikely to contain every possible IE found in that language. Because of this, evaluations containing IEs in both sets over-estimates the practical importance of tested methods.

To address this, we tested how well the approaches based on contextual word embeddings generalize to IEs outside the training set. For this experiment, we split our dataset into a training and testing set so that IEs from the testing set do not appear in the training set. Apart from this change, everything else remained the same as in section 4.3.1 above.

Since IEs in the test set are not present in the training set, the classification models cannot learn how to detect them based on word-data alone. We hypothesize that their detection is possible based on the contexts in which they appear. As the meaning of an IE is different from the literal meaning of its constituting words, it should appear in a different context. Neural networks with contextual word embeddings could detect such occurrences. Indeed, our results for the token- and sentence-level IE detection, presented in Tables 18 and 19, show that approaches that do not use contextual word embeddings fail to successfully detect IEs that did not occur in the training set. In contrast, MICE approaches using contextual embeddings extract useful information.

For token level results, shown in Table 18, due to the imbalanced class distribution, all approaches lag behind the default classifier concerning CA. For both the SVM baseline and MUMULS, this is the case also in terms of F_1 score. The MICE approach with ELMo and mBERT models manage to correctly classify many IEs. However, the results are worse than in the scenario where the same IEs are present in both the training and testing set. MICE with ELMO embeddings is again the best method, while CroSloEngual BERT is surprisingly unsuccessful.

Table 18: Comparison of results when classifying tokens and test set IEs are not present in the training set.

Method	CA	F_1 score
Default classifier	**0.903**	0.176
SVM baseline	0.870	0.029
MUMULS	0.873	0.000
MICE with Slovene ELMo	0.803	**0.866**
MICE with mBERT	0.733	0.803
MICE with CroSloEngual BERT	0.759	0.176

Sentence-level results in Table 19 show improved scores of all models, compared to token-level task. The SVM baseline and MUMULS still lag behind the default classifier concerning both CA and F_1 score. MICE approaches are better, with the Slovene ELMo variant achieving the best scores.

4.3.3 Cross-lingual evaluation of IEs

The results above show encouraging results for IE detection in a language with sufficiently large datasets. As recent research on cross-lingual embeddings shows that reasonably good transfer of trained models can be obtained for many tasks (Ruder et al., 2019; Artetxe & Schwenk, 2019; Robnik-Šikonja et al., 2020; Linhares Pontes et al., 2020), we attempt such a transfer of our models. We use the dataset from

Table 19: Comparison of results when classifying sentences and the test set IEs are not present in the training set.

Method	CA	F_1 score
Default classifier	0.828	0.906
SVM baseline	0.783	0.689
MUMULS	0.520	0.672
MICE with Slovene ELMo	**0.842**	**0.907**
MICE with mBERT	0.836	0.904
MICE with CroSloEngual BERT	0.771	0.837

the PARSEME shared task on automatic identification of verbal MWEs described in Section 4.2.2. We evaluated two contextual embeddings discussed in the previous sections: the Slovene ELMo embeddings and the multilingual BERT embeddings. We evaluated the cross-lingual MICE approach in the following manner:

- We evaluated MICE with Slovene ELMo embeddings on Slavic languages similar to Slovene, with datasets present in the PARSEME collection, i.e. Slovene, Croatian, and Polish. As the Slovene ELMo embeddings are not multilingual, they are unlikely to generalize to other languages. In future work, we plan to use these embeddings for prediction in other languages by using contextual cross-lingual mappings discussed in Section 2.
- We evaluated MICE with mBERT embeddings on all languages from the PARSEME collection. The mBERT model was trained on 104 languages, including every language present in the PARSEME dataset.

For both test-cases, we constructed balanced datasets, which consist of every sentence with IEs from the PARSEME dataset in a given language and an equal number of sentences without IEs, chosen at random from the same dataset. We evaluated the sentence-level classification task.

For the Slavic languages test, we trained the prediction model on the whole SloIE dataset, presented in Section 4.2.1. We did not train the model on any multilingual data to see whether the contextual embeddings alone are enough to generalize to other languages, at least to similar ones such as Croatian. For all PARSEME languages using MICE with mBERT, we split each dataset into the training, testing, and validation sets using a 60:30:10 ratio. We trained the model for each language on the training set and evaluated it on the testing set. For Slovene, Croatian, and Polish, we also trained MICE mBERT models on the SloIE dataset. The similarity of those languages means that additional data in the Slovene language could be beneficial. The results are presented in Table 20.

The monolingual evaluation results presented in Section 4.3.2 are also confirmed on the Slovene PARSEME dataset, as MICE with the Slovene ELMO model is capable of detecting idioms in that dataset. The same model generalizes very well to the PARSEME Croatian dataset, likely due to its similarity to Slovene. The generalization to Polish, which is a more distant Slavic language, is not successful. MICE models with mBERT also generalize well for a few languages. They obtain good results on Slovene and Croatian, likely due to the training on the SloIE corpus and generalization to similar Croatian idioms. The MICE mBERT models outperform default classifiers in French, Turkish, Lithuanian, Italian, Hebrew, and Basque, despite small amounts of training data, low numbers of IEs in training sets, most IEs only appearing once, and IEs in the testing set not appearing in the training set. They perform less well on other languages but are still capable of detecting some IEs.

MUMULS and the SVM baseline were both unable to detect IEs in other languages, obtaining the F_1 score of 0 in all cases.

4.4 Discussion on idiom detection

We showed that contextual word embeddings can be used with neural networks to successfully detect IEs in text. When contextual embeddings (ELMo or mBERT) were used as the first layer of a neural

Table 20: Results of the multilingual evaluation. The MICE models with Slovene ELMo embeddings were evaluated on Slavic languages similar to Slovene, while the variants with mBERT were tested for all languages in PARSEME dataset which contain IEs. We report F_1 scores and include the default classifier as a reference.

Language	Slovene ELMo	mBERT	Default F_1
Slovene	0.8163	0.8359	0.667
Croatian	0.9191	0.8970	0.667
Polish	0.2863	0.6987	0.667
English	-	0.650	0.667
French	-	0.814	0.667
German	-	0.622	0.667
Turkish	-	0.682	0.667
Romanian	-	0.625	0.667
Lithuanian	-	0.689	0.667
Italian	-	0.683	0.667
Hungarian	-	0.555	0.667
Hindi	-	0.562	0.667
Hebrew	-	0.693	0.667
Farsi	-	-	-
Basque	-	0.692	0.667
Spanish	-	0.340	0.667
Greek	-	0.484	0.667
Bulgarian	-	0.601	0.667

network with the same architecture as the existing MUMULS approach, we could obtain much better results. While the existing approaches performed well on the sentence-level classification of IEs present in the training set, they failed on token-level tasks and when detecting new IEs, not present in the training set. We showed that using fine-tuned contextual word embeddings allows the network to perform better on token-level classification and to successfully generalize to IEs that were not present in the training set. This opens an opportunity for the successful treatment of IEs in many downstream applications.

We evaluated our MICE approach on the SloIE dataset, a new, large dataset of Slovene idioms, as well as on the existing multilingual PARSEME datasets. SloIE dataset is larger than most existing datasets and should therefore be useful for further research into automatic idiom detection. Additionally, we evaluated how the size of the dataset affected the results and showed that our approaches perform well even when trained on smaller datasets.

We show that contextual word embeddings are capable of generalizing to other languages. When dealing with similar language pairs (e.g., Slovene-Croatian), both the monolingual ELMo embeddings and the multilingual BERT embeddings could detect idioms in Croatian text when trained only on Slovene. The multilingual BERT model detected idioms even in some more distant languages, though with reduced classification accuracy and F_1 scores.

Our work could be improved and extended in multiple ways. We only used embeddings that were pretrained on the general text and were not fine-tuned for the specific task of detecting idiomatic language. Several authors have shown (Li & Eisner, 2019; Devlin et al., 2019) that specializing embeddings for specific tasks can improve results on a variety of NLP tasks. Several such approaches could be applied to our task and would likely further improve the performance. Additionally, we intentionally used a simple network architecture that could be improved in the future. Finally, to put our models into practical use, we intend to apply MICE models in IE lexicon construction.

The work presented in Section 4 is described in full in (Škvorc et al., 2020), attached here as Appendix C.

5 Conclusions and further work

In this work, we tested and improved cross-lingual mappings of contextual embeddings and analyzed the cross-lingual transfer of trained models. We first presented several methods for cross-lingual mappings of contextual embeddings. The results show that ELMoVM and ELMoGAN methods enable successful cross-lingual mapping of ELMo contextual embeddings. The success largely depends on the dataset and successful fine-tuning of the methods' hyperparameters. Surprisingly, even low-quality contextual mapping dictionaries constructed from BabelNet and NER provide sufficient anchoring information if the resulting datasets are large enough.

We analyzed practically important issue of cross-lingual transfer of trained Twitter sentiment prediction models between languages. The results show a significant cross-lingual transfer potential using the models trained on similar languages, even in zero-shot transfer mode (without any data in the target language). The variant of multilingual BERT, CroSloEngual BERT, trained in T1.2, demonstrated excellent cross-lingual transfer between Croatian, Slovene, and English, with almost no loss. We demonstrated the use of contextual embeddings in monolingual and multilingual settings on the difficult linguistic problem of idiom detection. The proposed deep neural networks using either ELMo or BERT embeddings performed much better than existing approaches. Our MICE approach showed good cross-lingual transfer ability. It demonstrated that contextual word embeddings are capable of generalization to other languages, especially similar ones.

The proposed non-isomorphic mappings are very sensitive to hyper-parameter selection. In further work, we intend to work on a robust method to find hyper-parameters. We intend to test several more GAN architectures to find a more robust mapping.

We intend to expand our study of cross-lingual model transfer with additional BERT models, such as FinEst BERT, trained on English, Finnish and Estonian. Excellent performance of trilingual BERT models in cross-lingual transfer for sentiment prediction also requires further confirmations in other downstream tasks.

In our work with the idioms, we intend to test the contextual embeddings methodology on metaphors, where there is a similar problem of detecting different contexts. Metaphors are relevant for the creative language use in T5.3, where we plan to test the transformer neural architecture.

6 Associated outputs

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

Description	URL	Availability
ELMo embeddings	Clarin.si hdl.handle.net/11356/1277	Public (GPL v3)
CroSloEngual BERT embeddings	huggingface.co/EMBEDDIA/crosloengual-bert	Public(CC-BY 4.0)
Crosslingual NER	github.com/EMBEDDIA/crosslingual-NER	Public (GPL v3)
Vecmap changes	github.com/EMBEDDIA/vecmap-changes	Public (GPL v3)
Anchor point generation	github.com/EMBEDDIA/anchor-point-generation	Public (MIT)
SloIE idioms dataset	Clarin.si hdl.handle.net/11356/1335	Public (CC BY-NC-SA 4.0)
MICE source code	github.com/EMBEDDIA/MICE	Public (Apache)
ELMoGAN mapping method	github.com/EMBEDDIA/elmogan	Public (MIT)
SuPAR ELMo dependency parser	github.com/EMBEDDIA/supar-elmo	Public (GPL v3)

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. (2020). Cross-lingual alignments of ELMo contextual embeddings.	Draft	Appendix A
Robnik-Šikonja, M., Reba, K., Mozetič, I. (2020). Cross-lingual Transfer of Sentiment Classifiers. Submitted to *Slovenščina 2.0* journal.	Submitted	Appendix B
Škvorc, T., Gantar, A., Robnik-Šikonja, M. (2020). MICE: Mining Idioms with Contextual Embeddings. Submitted to *Knowledge Based Systems* journal. arXiv preprint arXiv:2008.05759 .	Submitted	Appendix C

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.

Ćavar, D., & Rončević, D. B. (2012). Riznica: the Croatian language corpus. *Prace filologiczne*, *63*, 51–65.

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.*

Constant, M., Eryiğit, G., Monti, J., Van Der Plas, L., Ramisch, C., Rosner, M., & Todirascu, A. (2017). Multiword expression processing: A survey. *Computational Linguistics*, *43*(4), 837–892.

Cook, P., Fazly, A., & Stevenson, S. (2008). The VNC-tokens dataset. In *Proceedings of the LREC workshop towards a shared task for multiword expressions (MWE 2008)* (pp. 19–22).

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).

Dozat, T., & Manning, C. D. (2017). Deep biaffine attention for neural dependency parsing. In *Proceedings of 5th international conference on learning representations, ICLR 2017.*

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).

Fu, Z., Xian, Y., Geng, S., Ge, Y., Wang, Y., Dong, X., ... de Melo, G. (2020). ABSent: Cross-lingual sentence representation mapping with bidirectional GANs. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 7756-7763.

Gantar, P., & Krek, S. (2011). Slovene lexical database. In *Natural language processing, multilinguality: 6th international conference* (pp. 72–80).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Hochreiter, S., & Schmidhuber, J. (1997, November). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780.

Jackendoff, R. (1997). *The architecture of the language faculty*. MIT Press.

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing (2nd edition)*. USA: Prentice-Hall, Inc.

Klyueva, N., Doucet, A., & Straka, M. (2017). Neural networks for multi-word expression detection. In *Proceedings of the 13th workshop on multiword expressions (MWE 2017)* (pp. 60–65).

Korkontzelos, I., Zesch, T., Zanzotto, F. M., & Biemann, C. (2013). Semeval-2013 task 5: Evaluating phrasal semantics. In *Second joint conference on lexical and computational semantics (* sem), volume 2: Proceedings of the seventh international workshop on semantic evaluation (semeval 2013)* (pp. 39–47).

Krek, S., Gantar, P., Holdt, Š. A., & Gorjanc, V. (2016). Nadgradnja korpusov Gigafida, Kres, ccGigafida in ccKres. In *Proceedings of Language technologies and digital humanistics* (pp. 200–202).

Li, X. L., & Eisner, J. (2019). Specializing word embeddings (for parsing) by information bottleneck. 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. 2744–2754).

Linhares Pontes, E., Moreno, J. G., & Doucet, A. (2020). Linking named entities across languages using multilingual word embeddings. In *20th ACM/IEEE joint conference on digital libraries, JCDL 2020.*

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.*

Ljubešić, N., & Erjavec, T. (2011). hrWaC and slWaC: Compiling web corpora for Croatian and Slovene. In *International conference on text, speech and dialogue* (pp. 395–402).

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

Mozetič, I., Grčar, M., & Smailović, J. (2016). Multilingual Twitter sentiment classification: The role of human annotators. *PLOS ONE*, *11*(5).

Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217-250.

Nivre, J., Abrams, M., & Agić, Ž. (2020). *Universal Dependencies 2.6.* Retrieved from `http://hdl.handle.net/11234/1-2988`

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).

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).

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th annual meeting of the association for computational linguistics: System demonstrations.*

Robnik-Šikonja, M., Reba, K., & Mozetič, I. (2020). Cross-lingual transfer of Twitter sentiment models using a common vector space. In *Proceedings of language technologies & digital humanities.*

Robnik-Šikonja, M., Reba, K., & Mozetič, I. (2020). *Cross-lingual transfer of sentiment classifiers.* (submitted)

Ruder, S., Vulić, I., & Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, *65*, 569–631.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., & Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. In *International conference on intelligent text processing and computational linguistics* (pp. 1–15).

Savary, A., Ramisch, C., Cordeiro, S., Sangati, F., Vincze, V., QasemiZadeh, B., ... Doucet, A. (2017). The PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the 13th workshop on multiword expressions (MWE 2017)* (pp. 31–47).

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*.

Škvorc, T., Gantar, P., & Robnik-Šikonja, M. (2020). *MICE: Mining idioms with contextual embeddings.* arXiv preprint 2008.05759. (submitted)

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.

Ulčar, M., & Robnik-Šikonja, M. (2020). *Cross-lingual alignments of ELMo contextual embeddings.* (draft)

Ulčar, M., & Robnik-Šikonja, M. (2020a). High quality ELMo embeddings for seven less-resourced languages. In *Proceedings of the 12th Language resources and evaluation conference, LREC 2020* (pp. 4733–4740).

Ulčar, M., & Robnik-Šikonja, M. (2020). FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In *Proceedings of Text, Speech, and Dialogue, TSD 2020* (p. 104-111).

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).

Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., ... Pyysalo, S. (2019). *Multilingual is not enough: BERT for Finnish.* arXiv preprint arXiv:1912.07076.

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).

Wehrmann, J., Becker, W., Cagnini, H. E., & Barros, R. C. (2017). A character-based convolutional neural network for language-agnostic Twitter sentiment analysis. In *2017 international joint conference on neural networks (IJCNN)* (pp. 2384–2391).

Yang, Z., Chen, W., Wang, F., & Xu, B. (2018). Improving neural machine translation with conditional sequence generative adversarial nets. 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. 1346–1355).

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., ... Hsieh, C.-J. (2020). Large batch optimization for deep learning: Training BERT in 76 minutes. In *International conference on learning representations, ICLR 2019.*

Appendix A: Cross-lingual alignments of ELMo contextual embedding

Cross-lingual alignments of ELMo contextual embeddings

Matej Ulčar and Marko Robnik-Šikonja

University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, Ljubljana, Slovenia
{matej.ulcar, marko.robnik}@fri.uni-lj.si

Abstract. Building machine learning prediction models for a specific NLP task requires sufficient training data, which can be difficult to obtain for low-resource languages. Cross-lingual embeddings map word embeddings from a low-resource language to a high-resource language so that a prediction model trained on data from the high-resource language can also be used in the low-resource language. To produce cross-lingual mappings of recent contextual embeddings, anchor points between the embedding spaces have to be words in the same context. We address this issue with a new method for creating datasets for cross-lingual contextual alignments. Based on that, we propose novel cross-lingual mapping methods for ELMo embeddings. Our linear mapping methods use existing vecmap and MUSE alignments on contextual ELMo embeddings. Our new nonlinear ELMoGAN mapping method is based on GANs and does not assume isomorphic embedding spaces. We evaluate the proposed mapping methods on nine languages, using two downstream tasks, NER and dependency parsing. The ELMoGAN method performs well on the NER task, with low cross-lingual loss compared to direct training on some languages. In the dependency parsing, linear alignment variants are more successful.

Keywords: contextual embeddings, ELMo, GAN, cross-lingual models, non-linear vector alignment, non-isomorphic cross-lingual alignment, vecmap, MUSE

1 Introduction

Word embeddings are representations of words in a numerical form, as vectors of typically several hundred dimensions. The vectors are used as input to machine learning models; these are generally deep neural networks for complex language processing tasks. The embedding vectors are obtained from specialized neural network-based embedding algorithms. The quality of embeddings depends on the amount of semantic information expressed in the embedded space through distances and directions. For that reason, static pre-trained word embeddings, such as word2vec [25] or fastText [8], have in large part been recently replaced by contextual embeddings, such as ELMo [31] and BERT [11].

Contextual embeddings generate a different word vector for the same word for every context it appears in. BERT models and their derivatives are most often used as a closed system, where the entire model is fine-tuned on a downstream task. ELMo models, on the other hand, generate word vectors for each word occurrence, and these vectors are then used for training various NLP models. The ELMo neural network model consists of three layers of neurons. Embeddings are typically a concatenation of network weights in all three layers. BERT models consist of 12 or 24 layers, and vector extraction typically uses a combination of the last four layers. For BERT, this approach may lose a lot of information; therefore, the extracted BERT vectors are rarely used and are less successful than ELMo vectors, see, e.g., [38]. A smaller size of ELMo models, compared to BERT, also offers better explainability of the end-task models.

Modern word embedding spaces exhibit similar structures across languages, even when considering distant language pairs like English and Vietnamese [25]. This means that embeddings independently produced from monolingual text resources can be aligned, resulting in a common cross-lingual representation, called cross-lingual embeddings, which allows for fast and effective integration of information in different languages. For low-resource languages, training NLP models can be difficult because of a lack of data for a specific task. The idea of cross-lingual alignment is to use an already existing model trained on a high-resource language and map the word embeddings from a low-resource language vector space to the high-resource language vector space. That way, the words with the same meaning in both languages have very similar vectors, which is not the case before the alignment procedure.

Cross-lingual approaches can be sorted into several groups. The first group of methods uses monolingual embeddings with (an optional) help from bilingual dictionaries to align the embeddings. These methods are mostly used for static embeddings, such as word2vec and fastText. The second group of approaches uses bilingually aligned (comparable or even parallel) corpora for joint construction of embeddings in all involved languages. The third type of approach is based on large pretrained multilingual masked language models such as BERT [11]. In this work, we present an extension of the first group of approaches to contextual embeddings. We focus on improvements of cross-lingual mappings for ELMo contextual embeddings. Currently, the most successful alignment methods assume that the embedding spaces in different languages are isomorphic [6, 9], which is generally not the case. Researchers have observed that the monolingual embedding spaces of two different languages are not completely isomorphic, which is especially true for distant languages [30, 39]. As a result, many of these methods are unstable or unsuccessful when confronted with distant language pairs.

We propose novel methods for isomorphic and non-isomorphic alignment of contextual embeddings, such as ELMo. For that purpose, we first construct novel contextual mapping datasets based on parallel corpora and dictionaries. In a novel non-isomorphic approach, we use generative adversarial networks (GANs) [16], that produce nonlinear mappings between the embedding spaces.

The main contributions of this work are i) a novel approach to create datasets needed in the cross-lingual alignment of contextual embeddings, ii) novel isomorphic and non-isomorphic cross-lingual alignments of ELMo embeddings, iii) their evaluation on nine low-resource languages and two downstream tasks, named entity recognition (NER) and dependency parsing (DP). The results show the successful cross-lingual transfer of tested approaches. The best alignment method is dependent on the task.

The paper is split into four further sections. In Section 2, we present the background on cross-lingual alignment and ELMo and cover related work on cross-lingual embeddings. The construction of special datasets used for training the alignments of contextual embeddings is presented in Section 3. In Section 4, we describe the proposed ELMoGAN alignment method. In Section 5, we evaluate the proposed alignment methods on two downstream tasks. We summarize our work in Section 6 and discuss opportunities for further work.

2 Background and related work

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. [40] present a detailed overview and classification of cross-lingual methods.

In Section 2.1, we describe how two monolingual embedding spaces can be aligned with the optional help from a bilingual dictionary. This work's main focus is extending existing approaches that work with non-contextual embeddings to contextual ELMo embeddings. For this reason, we present the background on ELMo contextual embeddings in Section 2.2. The related work on non-contextual mappings is given in Section 2.3, and on contextual mapping in Section 2.4.

2.1 Alignment of monolingual embeddings

Cross-lingual alignment methods take precomputed word embeddings for each language and align them with the 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 works by Artetxe et al. [5].

The open source implementation of the method described by Artetxe et al. [6, 5], named *vecmap*[1], is able to align monolingual embeddings either using supervised, semi-supervised or unsupervised approach.

The supervised approach requires a large bilingual dictionary, which is used to match embeddings of the 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 [6].

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 the 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, entries of a bilingual dictionary can be used as anchors for the alignment map for supervised and semi-supervised approaches. Lately, researchers have proposed approaches that do not require the use of a bilingual dictionary but rely on an adversarial approach [9] or use the frequencies of the words [6] to find a required transformation. These are called unsupervised approaches.

The Facebook research project MUSE[2] can find a cross-lingual map with the use of a bilingual dictionary (supervised) or without one (unsupervised approach). The unsupervised approach works by using adversarial training to find the starting linear mapping. A synthetic dictionary is extracted from this mapping, which is used to fine-tune the starting mapping using the Procrustes approach, described in detail by Conneau et al. [9].

2.2 ELMo contextual embeddings

ELMo (Embeddings from Language Models) embedding [31] 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 [18] . In the first LSTM, we try to predict the following word, based on the given past words, where the embeddings from the CNN layer represent each word. In the second LSTM, we try to predict the preceding

[1] `https://github.com/artetxem/vecmap`
[2] `https://github.com/facebookresearch/MUSE`

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 [31]. To train the ELMo network, one puts one sentence at a time on the input. 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. Still, 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

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

where L is the number of layers (ELMo uses $L = 2$), index j refers to the 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 [31]. Later, BERT models turned out to be even more successful on these tasks. However, concerning the quality of extracted vectors, ELMo can be advantageous as its information is condensed in only three layers. In comparison, the information in multilingual BERT is scattered over 14 layers.

2.3 Related work on non-contextual mappings

Cross-lingual alignment methods align precomputed monolingual word embeddings from two or more languages. The word vectors from all the languages are mapped into a common vector space. This can be the same vector space as one of the original monolingual embeddings or a completely independent vector space. These methods aim to represent the words with the same meaning in different languages with as similar vectors as possible. Concerning the data used, the alignment methods can be split into supervised and unsupervised methods. Supervised methods determine the alignment of the embeddings with the use of bilingual dictionaries. Unsupervised methods do not use any bilingual data. Conneau et al. [10] trained the unsupervised alignment using adversarial training. Artetxe et al. [6] first constructed a low-quality seed dictionary using the assumption that the two vector spaces are isometric and then iteratively updated the mapping and dictionary until convergence.

Artetxe et al. [5] comprehensively summarize existing linear methods, showing that the state-of-the-art linear alignment methods can be summarized as an orthogonal mapping. The difference between various methods is solely due to different approaches to vector manipulation (such as normalization, whitening, etc.) before the mapping extraction.

Nakashole and Flauger [27] show that, in a small neighborhood, linear mapping methods work well; however, the linearity assumption does not hold in general, especially for distant languages [30]. A few nonlinear alignment methods have been proposed. Lu et al. [24] trained nonlinear mapping using Deep Canonical Correlation Analysis (DCCA) [4], which is an expanded version of a linear Canonical Correlation Analysis (CCA) method, using deep neural networks. They showed that DCCA performs better than linear CCA. Recently, Zhao and Gilman [44] proposed a nonlinear mapping method, using kernel CCA (KCCA). KCCA projects the vectors into a higher dimensional space and then performs CCA in the new vector space. Zhao and Gilman [44] report that DCCA has to fine-tune many hyper-parameters and show that KCCA outperforms both DCCA and CCA, especially when data is scarce. In contrast, Lu et al. [24] observe that DCCA scales better with data size than KCCA.

Conneau et al. [10] used the adversarial training based on generative adversarial networks (GAN) to train a linear mapping between vector spaces. Yang et al. [43] have used full GAN models for neural machine translation. Fu et al. [15] trained a bidirectional GAN for cross-lingual alignment of sentence embeddings, improving the results over linear and nonlinear state-of-the-art-methods on sentence alignment task.

2.4 Related work on contextual mappings

All the above work only concerned static embeddings, not dynamic, contextual embeddings. Schuster et al. [36] produced cross-lingual alignments of contextual ELMo embeddings. While each occurrence of a word in contextual embeddings is represented by a different vector, Schuster et al. [36] hypothesized that in these vectors form clusters. Based on this assumption, they assigned each word a single static vector by calculating the average vector over all word occurrences in a large corpus. They used a linear MUSE method to calculate the alignments of the averaged vectors. This approach's problem is the assumption of isomorphic spaces and loss of information if this assumption is not true in the local context.

Aldarmaki and Diab [3] used parallel corpora to produce the embedding vectors. They aligned the corpora on the word level, using Fast Align [14], calculated the ELMo embeddings on the aligned corpora, and extracted a dictionary from the word-level alignments. Their approach showed good results in a sentence translation retrieval task. 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. They applied their approach to three languages (English, German, Spanish). This approach is similar to the linear mappings applied to ELMo, which we describe in Section 4.3. The difference is that we use much larger dictionaries and test on many more language pairs.

3 Datasets for alignment of contextual embeddings

This section explains how we generated training datasets required for cross-lingual alignment of contextual ELMo embeddings. We also present the language resources we used in the creation of the dataset.

Supervised cross-lingual vector alignment approaches assume a bilingual dictionary is provided, where each word from the dictionary has its own embedding vector. For static, non-contextual embeddings this is straight-forward. For contextual embeddings, the word vector depends on the context the word appears in. For every context, the word gets a different vector. Schuster et al. [36] solved this by averaging all the vectors of a given word, as described in Section 2. Some information is lost when using this approach, as words have multiple meanings. For example, the word "bark" can refer to the sound a dog makes, a sailing boat, or the outer part of a tree trunk. Furthermore, two meanings may be represented with one word in one language but with two different words in another language.

We solve this issue by separately aligning each occurrence of a word. We start with a parallel corpus, aligned on a paragraph-level to have matching contexts in two languages. Let i indicate the index of a context from a parallel corpus P. Let A and B represent the first and the second language in a language pair. Then P_i^A is the i-th paragraph/context from corpus P in language A. Given a bilingual dictionary D, let j indicate the index of a word pair in the dictionary so that the dictionary is composed of pairs $(D_j^A, D_j^B), \forall j \in \|D\|$.

We construct our dataset by parsing the parallel corpus. For each word $a \in P_i^A$, we check whether its lemma appears in D^A. If it does, given its dictionary index j, we check whether D_j^B is a lemma of any word from P_i^B. If it is, we add a tuple $(iD_j^A, iD_j^B, e(iD_j^A, P_i^A), e(iD_j^B, P_i^B))$ to our dataset, where $e(iD_j^A, P_i^A)$ and $e(iD_j^B, P_i^B))$ are ELMo embeddings of the two dictionary words iD_j^A and iD_j^B, computed in the context P_i for each of the languages, respectively. We considered at most 20 different contexts of each lemma to not overwhelm the dataset with frequent words (such as stop words). For lemmatization of the corpora, we used the Stanza tool [34]. Note that we only used lemmatized corpora for dictionary look-up; for generating the embedding, we used the non-lemmatized corpora.

As we explained in Section 2.2, ELMo models are deep neural networks with three hidden layers. The first layer is non-contextual CNN, followed by two contextual biLSTM layers. The final embedding vectors are constructed from vectors of all three layers, The first vector is contextually independent, while the second and third are contextually dependent. In our cross-lingual alignment approaches for ELMo, we align vectors from each of the three layers separately. Thus, we need a separate dataset for each layer. We created two such contextual datasets for each language pair, one for each of the contextual ELMo layers. For the non-contextual ELMo layer, we produce embeddings for every word pair in the bilingual dictionary. As the non-contextual ELMo vectors are the same for all word contexts, the size of that dataset is identical to the bilingual dictionary size.

We split the created datasets into a training and evaluation part. We separately split data for each language pair and each ELMo layer. The train part

has 98.5% of word vector pairs, and the evaluation part has 1.5% of word vector pairs.

In our work, we considered eleven language pairs from nine different languages. The language pairs along with the sizes of bilingual dictionaries, parallel corpora, and the final training dataset are presented in Table 1. For English, we used the original English 5.5B ELMo model[3]. For Russian, we used the ELMo model trained by DeepPavlov[4] on the Russian WMT News. For other languages, we used ELMo models trained by Ulčar and Robnik-Šikonja [42].

Table 1: The sizes of dictionaries and parallel corpora used in the creation of a dataset for contextual mappings, as well as the size of the resulting dataset for alignment of ELMo embeddings. The sizes of dictionaries are reported in the number of word pairs, the sizes of parallel corpora in the number of matching contexts, and the sizes of resulting datasets in the number of matched words in matched sentence pairs. The Type column describes the dictionary creation approach: "direct" means that the dictionary was created directly from wiktionary, "triang" means that the dictionary was created from wiktionary using triangulation via English, and "OES" stands for the Oxford English-Slovene dictionary.

Language pair	Type	Dictionary	Parallel corpus	ELMo dataset
English-Estonian	direct	11 022	12 486 898	77 800
English-Finnish	direct	89 307	27 281 566	283 000
English-Croatian	direct	3448	35 131 729	44 800
English-Lithuanian	direct	13 960	1 415 961	62 800
English-Latvian	direct	10 224	519 553	43 800
English-Russian	direct	103 850	25 910 105	363 800
English-Slovenian	direct	9634	19 641 457	89 800
English-Slovenian	OES	182 787	19 641 457	294 318
English-Swedish	direct	51 961	17 660 152	270 000
Estonian-Finnish	direct	2191	9 504 879	12 800
Estonian-Finnish	triang	43 313	9 504 879	78 200
Croatian-Slovenian	direct	266	15 636 933	3400
Croatian-Slovenian	triang	3669	15 636 933	31 600
Lithuanian-Latvian	direct	2478	219 617	11 200
Lithuanian-Latvian	triang	14 545	219 617	28 200

We used the OpenSubtitles parallel corpora[5] [22] from the Opus web page[6] for each pair of languages. The dictionaries we used are bilingual dictionaries extracted from wiktionary, using wikt2dict[7] tool [1]. The tool allows for direct dictionary extraction, as well as triangulation via a third language. In the tri-

[3] https://allennlp.org/elmo
[4] https://github.com/deepmipt/DeepPavlov
[5] https://www.opensubtitles.org/
[6] http://opus.nlpl.eu
[7] https://github.com/juditacs/wikt2dict

angulation case, given three languages, A, B and C, we construct a bilingual dictionary for languages A and B, so that for every word $a \in A$, we find its translation $c \in C$ from $A - C$ dictionary. We then search for the translation of the word c in language B in the $C - B$ dictionary. We label this translated word b. The dictionary created using triangulation consists of pairs $a - b$.

The dictionaries made using the wikt2dict tool are noisy, so we manually filtered them. We replaced the accented vowels with their non-accented variants in languages that do not use accented letters for vowels (e.g., Slovene and Russian). We removed the extra non-alphabetic characters, such as hash symbol, brackets, pipe, etc. We also removed all the entries which contained multiple-word terms. We leave the extension to the alignment of multi-word terms to further work.

We used direct bilingual dictionaries for all language pairs, where one of the languages was English. We used direct dictionaries and dictionaries created with triangulation via English for all the pairs where both languages are not English. For the English-Slovene pair, we also used a large, high quality, handmade proprietary Oxford English-Slovene dictionary.

4 Contextual alignments

This section first describes our proposed ELMoGAN method for nonlinear cross-lingual alignment of contextual ELMo embeddings. In Section 4.1, we describe the architecture of our model, and in Section 4.2, we present training of the contextual alignments. Based on the constructed contextual alignment datasets, it is also possible to align contextual embeddings with classical linear mappings. We describe this approach in Section 4.3.

4.1 Architecture of ELMoGAN

Generative Adversarial Networks (GANs) [16] consist of two connected neural models, a generator and a discriminator. The two models are trained simultaneously via an adversarial process. The discriminator attempts to discern whether the data passed to its input is real or fake (i.e. artificially generated). At the same time, the generator attempts to generate artificial data, which can fool the discriminator. GANs play a zero-sum game, where the discriminator's success means the generator's failure and vice versa. By simultaneously training both networks, they both improve. GANs are mostly used on images, where the described process can lead to compelling new generated images.

Following the success of GANs in neural machine translation [43] and cross-lingual embeddings alignment [9, 15], we propose a novel supervised nonlinear mapping method using bidirectional GANs. We based our contextual alignment method, called ELMoGAN, on the model of Fu et al. [15]. Contrary to Fu et al. [15], who only used their method with non-contextual fastText embeddings [8] to align sentences, we align contextual ELMo embeddings [31], which is only possible by constructing a special contextual mapping datasets, described in Section 3.

As illustrated in Figure 1, the mapping GAN comprises the generator module and discriminator module. The generator module contains two generators that map vectors from one vector space to the other. Specifically, for a pair of languages L_1 and L_2, one generator will map from L_1 to L_2, and the second will map from L_2 to L_1. The two generators are completely independent of one another, and they do not share the data during training. The discriminator module contains two discriminators. The first discriminator tries to predict whether a given pair of vectors represent the same token, i.e. if the first vector represents the word x in L_1 and the second vector represents the translation of the word x in L_2. The second discriminator attempts to learn the difference between the direction of mapping. For a given pair of vectors, it predicts whether they are a vector from L_1 and its mapping to L_2 or a vector from L_2 and its mapping to L_1.

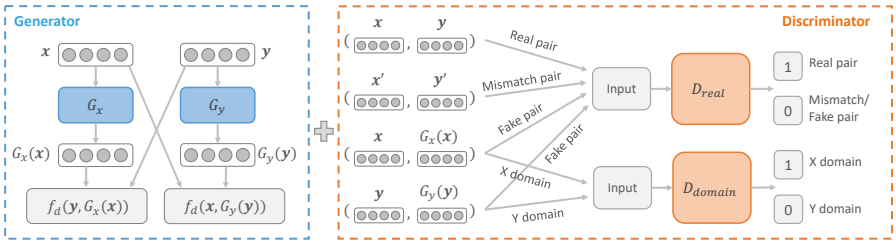


Fig. 1: The schema of the GAN, proposed by Fu et al. [15] for sentence alignment. The image is taken from that source.

Compared to the ABSent model by Fu et al. [15], in ELMoGAN, we increased the sizes of all the hidden layers in generators and discriminators. We also significantly lowered the learning rate as we achieved poor results with the learning rate used by Fu et al. [15]. Both generators in ELMoGAN have the same architecture: the input layer is followed by three fully connected feed-forward layers of 2048, 4096, and 2048 nodes. We used the ReLU activation function for all three layers. We added a batch normalization layer between each fully connected layer. The output layer has the same size as the input layer. It uses hyperbolic tangent as the activation function so that the output is between -1 and $+1$. Both discriminators also have the same architecture. We first concatenate the two input vectors, then feed them to three consecutive fully connected feed-forward layers with leaky ReLU ($\alpha = 0.2$). The output layer is a single neuron with the sigmoid activation.

4.2 Training of ELMoGAN

We jointly trained the generator and discriminator modules using the parallel ELMo vectors datasets, described in Section 3. We trained ELMoGAN with the

batch size of 256, Adam optimizer with learning rate 2×10^{-5}, and learning rate decay 10^{-5}. For each language pair, we trained three mapping models, one for each of the ELMo layers. For all three models, we used the same settings. We feed the generators the word vectors from our training dataset. On the input of the first generator are the vectors from L_1, and on the output, there are the matching vectors from L_2; vice-versa is true for the second generator. We feed the first discriminator various pairs of vectors; some represent the same token (True), others represent two different tokens or no token at all (False). For the vector pairs labeled as True, we take the matching pairs from our train dataset. For the vector pairs labeled as False, we have three types of pairs. The first type is two randomly selected vectors from our dataset (one from each vector space). The second type is vectors from L_1 and their mappings, using generator one. The third type is vectors from L_2 and their mappings, using generator two.

We produced two different versions of the ELMoGAN, based on the number of iterations the model was trained for. The first version (ELMoGAN-10k) was trained for a fixed number of 10 000 iterations for each layer of each language pair. For the second version (ELMoGAN-O), we trained models with several different numbers of iterations. We evaluated them on a dictionary induction task and selected the number of iterations that gave the best result. The details of selecting the number of iterations are presented in Appendix A.

4.3 Cross-lingual linear mappings for contextual embeddings

It is possible to compute cross-lingual mappings between contextual embeddings based on the standard assumption that the aligned spaces are largely isomorphic. Below, we shortly describe methods belonging to this type of alignment approach.

With a large enough collection of words in matching contexts (as described in Section 3), we compute their contextual embedding vectors and align them with any of the non-contextual mapping methods. We use two such methods, the vecmap library [5], which aligns both embedding spaces, and the MUSE library [9], which only aligns target vectors and is therefore computationally more efficient. As discussed in Section 2.4, a similar approach was proposed by Aldarmaki and Diab [3] but did not use large contextual datasets based on high-quality dictionaries as we did.

5 Evaluation

We evaluated the ELMoGAN-10k and ELMoGAN-O methods, trained as described in Section 4.2 on two downstream tasks, named entity recognition and dependency parsing. The results are presented separately for each task in Sections 5.1 and 5.2. We compare these methods with two linear mapping methods, MUSE [10] and Vecmap [6, 5], adapted for contextual embeddings, as described in Section 4.3. These are linear cross-lingual mapping methods that assume isomorphic translation between vector spaces. For training of alignments, we

used the same datasets, described in Section 3. In all of our experiments, we use embeddings mapping from an evaluation to train language, i.e. we map the embeddings of the language used for the evaluation to the vector space of the language, which was used during the training of the model.

To better interpret the obtained results, we conducted two further ablation studies. In Figure 3, we tested the importance of alignment dataset size. We used the English-Slovene pair, where we have available a large, high-quality proprietary Oxford English-Slovene dictionary, instead of publicly available wiktionary. In Section 5.4, we tested different variants of the Vecmap alignment approach to check if we can avoid transforming both the source and target vector space and thereby significantly speed-up the approach.

5.1 Named Entity Recognition

Named entity recognition (NER) is an information extraction task that seeks to locate and classify named entities mentioned in unstructured text into predefined categories such as the person names, organizations, locations, medical codes, time expressions, quantities, monetary values, etc. The labels in the used NER datasets are simplified to a common label set of four labels present in all the addressed working languages. These labels are a person, location, organization, and other. The latter encompasses all named entities that do not fall in one of the three mentioned classes and all the tokens that are not named entities. The datasets used in the evaluation on the NER task are shown in Table 2, along with some basic statistics of the datasets.

Table 2: The collected datasets for NER task and their properties: number of sentences, number of tagged words, availability, and link to the corpus location).

Language	Corpus	Sentences	Tags	Avail.	Location
Croatian	hr500k [23]	25000	29000	public	link
English	CoNLL-2003 NER [41]	21000	44000	public	link
Estonian	Estonian NER corpus [21]	14000	21000	public	link
Finnish	FiNER data [35]	14500	17000	public	link
Latvian	LV Tagger train data	10000	11500	public	link
Lithuanian	TildeNER	5476	7024	limited	NA
Slovene	ssj500k [20]	9500	9500	public	link
Swedish	Swedish NER	8500	7500	public	link

We present the results using the Macro F_1 score, which is an average of F_1 scores for each class we are trying to predict, excluding the class Other (i.e. not a named entity).

The upper part of Table 3 shows a typical cross-lingual transfer learning scenario, where the model is transferred from resource-rich language (English) to less-resourced languages. In this case, the non-isomorphic ELMoGAN methods, in particular the ELMoGAN-10k variant, are superior to isomorphic Vecmap

Table 3: Comparison of different methods for cross-lingual mapping of contextual ELMo embeddings, evaluated on the NER task. The best Macro F_1 score for each language pair is in bold. The "Reference" column represents a direct learning on the target language without cross-lingual transfer. The upper part of the table contains a scenario of cross-lingual transfer from English to a less-resourced language, and lower part of the table shows a transfer between similar languages.

Source.	Target.	Dictionary	Vecmap	ELMoGAN-O	ELMoGAN-10k	MUSE	Reference
English	Croatian	direct	**0.385**	0.279	0.345	0.024	0.810
English	Estonian	direct	0.554	0.682	**0.737**	0.284	0.895
English	Finnish	direct	0.672	0.708	**0.788**	0.229	0.922
English	Latvian	direct	0.499	**0.650**	0.630	0.216	0.818
English	Lithuanian	direct	0.498	0.476	**0.575**	0.208	0.755
English	Slovenian	direct	0.548	0.588	**0.664**	0.060	0.850
English	Swedish	direct	0.786	0.686	**0.797**	0.568	0.852
Croatian	Slovenian	direct	0.387	0.279	0.250	**0.418**	0.850
Croatian	Slovenian	triang	**0.731**	0.365	0.420	0.592	0.850
Estonian	Finnish	direct	**0.517**	0.288	0.302	0.278	0.922
Estonian	Finnish	triang	**0.779**	0.705	0.677	0.296	0.922
Finnish	Estonian	direct	0.477	0.263	0.331	**0.506**	0.895
Finnish	Estonian	triang	0.581	0.563	**0.595**	0.549	0.895
Latvian	Lithuanian	direct	**0.423**	0.376	0.367	0.345	0.755
Latvian	Lithuanian	triang	0.569	0.632	**0.637**	0.378	0.755
Lithuanian	Latvian	direct	0.263	0.305	0.318	**0.604**	0.818
Lithuanian	Latvian	triang	0.359	0.691	**0.713**	0.710	0.818
Slovenian	Croatian	direct	0.361	0.260	0.328	**0.485**	0.810
Slovenian	Croatian	triang	**0.566**	0.490	0.427	0.518	0.810

and MUSE approaches. In this scenario, ELMoGAN-10k is always the best or close to the best mapping approach. This is not always the case in the lower part of Table 3, which shows the second most important cross-lingual transfer scenario: transfer between similar languages. In this scenario, ELMoGAN is the best in three language pairs. Isomorphic Vecmap and MUSE perform best in nine language pairs (five times Vecmap and four times MUSE). We hypothesize that the reason for isomorphic mappings' better performance is the similarity of tested language pairs and less violation of isomorphism assumption the Vecmap and MUSE method make. The results of the MUSE method support this hypothesis. While MUSE performs worst in most cases of transfer from English, the performance gap is smaller for transfer between similar languages. MUSE is sometimes the best method for similar languages, but the results of MUSE fluctuate greatly between language pairs. The second possible factor explaining the results is the quality of the dictionaries, which are in general better for combinations involving English. In particular, dictionaries obtained by triangulation via English are of poor quality, and non-isomorphic transformation might be more affected by imprecise anchor points.

In general, even the best cross-lingual prediction models lag behind the reference model without cross-lingual transfer. The differences in Macro F_1 score

are small for some languages (e.g., 5.5% for English-Swedish), but they are significantly larger for most of the languages.

5.2 Dependency parsing

Dependency parsing task (DP) constructs a dependency tree of a given sentence. In DP, all the words in a sentence are arranged into a hierarchical tree based on their semantic dependencies. Each word has at most one parent node, and only the root word has no parent. A word can have multiple children nodes. In addition to predicting the tree's structure, the task is also to label the hierarchical dependencies.

Table 4: Dependency parsing datasets and their properties: the treebank, number of sentences, number of tokens, and information about the size of the split.

Language	Treebank	Tokens	Sentences	Train	Validation	Test
Croatian	SET [2]	199409	9010	6914	960	1136
English	EWT [37]	254855	16622	12543	2002	2077
Estonian	EDT [26]	438171	30972	24633	3125	3214
Finnish	TDT [17, 33]	202697	15135	12216	1364	1555
Latvian	LVTB [32]	220536	13643	10156	1664	1823
Lithuanian	ALKSNIS [7]	70051	3642	2341	617	684
Russian	GSD	98000	5030	3850	579	601
Slovene	SSJ [12, 20]	140670	8000	6478	734	788
Swedish	Talbanken [28]	96858	6026	4303	504	1219

As the dependency parsing architecture, we use the SuPar tool by Yu Zhang[8], which is based on the deep biaffine attention [13]. We modified the SuPar tool to accept ELMo embeddings on the input; specifically, we used the concatenation of the three ELMo layers. We made the modified code publicly available[9]. We trained the parser for 10 epochs, using datasets in nine languages (Croatian, English, Estonian, Finnish, Latvian, Lithuanian, Russian, Slovene, and Swedish). The datasets are obtained from the Universal Dependencies [29] version 2.3. The datasets used and their basic statistics are shown in Table 4.

We used two evaluation metrics in the dependency parsing task, the unlabelled and labeled attachment scores (UAS and LAS) on the test set. The UAS and LAS are standard accuracy metrics in DP. The UAS score is defined as the proportion of tokens that are assigned the correct syntactic head. The LAS score is the proportion of tokens assigned the correct syntactic head and the correct dependency label [19].

The Vecmap mapping method outperforms both ELMoGAN methods on all language pairs in this task. Larger dictionaries, created with triangulation,

[8] `https://github.com/yzhangcs/parser`
[9] `https://github.com/MatejUlcar/parser/tree/elmo`

Table 5: Comparison of different contextual cross-lingual mapping methods on dependency parsing task. Results are reported as unlabeled attachments score (UAS) and labeled attachment score (LAS).The column "Direct" stands for direct learning on the target (i.e. evaluation) language without cross-lingual transfer. The languages are represented with their international language codes ISO 639-1.

Train lang.	Eval. lang.	Dict.	Vecmap UAS	Vecmap LAS	ELMoGAN-O UAS	ELMoGAN-O LAS	ELMOGAN-10k UAS	ELMOGAN-10k LAS	MUSE UAS	MUSE LAS	Direct UAS	Direct LAS
en	hr	direct	**73.96**	**60.53**	69.75	50.20	65.66	38.95	71.01	54.89	91.74	85.84
en	et	direct	**62.08**	**40.62**	52.75	30.05	43.48	22.97	58.76	34.07	89.54	85.45
en	fi	direct	**64.40**	**45.32**	49.41	29.35	42.54	22.69	55.03	37.61	90.83	86.86
en	lv	direct	**77.84**	**65.97**	68.43	46.09	67.30	38.38	76.26	63.45	88.85	82.82
en	lt	direct	**67.92**	**39.62**	56.60	30.19	62.26	24.53	66.04	37.74	55.05	24.39
en	ru	direct	**72.00**	**16.62**	66.46	9.23	61.85	8.31	/	/	89.33	83.54
en	sl	direct	**79.01**	**59.84**	68.38	48.87	64.98	44.86	77.18	56.53	93.70	91.39
en	sv	direct	82.08	72.74	74.45	60.39	75.14	60.69	**82.17**	**72.78**	89.70	85.07
hr	sl	direct	**85.47**	**72.70**	54.06	34.17	55.34	32.77	83.45	69.08	93.70	91.39
hr	sl	triang	**87.70**	**76.51**	73.23	60.95	70.86	54.62	**87.70**	76.40	93.70	91.39
et	fi	direct	**79.14**	**66.09**	52.97	34.25	49.68	28.37	76.66	60.01	90.83	86.86
et	fi	triang	**80.94**	**67.35**	54.91	31.94	54.40	26.91	76.96	63.37	90.83	86.86
fi	et	direct	**75.81**	57.32	54.23	34.19	54.64	32.90	74.96	**58.14**	89.54	85.45
fi	et	triang	**79.04**	**61.86**	61.19	39.46	56.41	32.58	76.74	60.27	89.54	85.45
lv	lt	direct	**72.38**	**51.43**	64.76	45.71	61.90	35.24	67.62	50.48	55.05	24.39
lv	lt	triang	**75.24**	50.48	68.57	39.05	69.52	34.29	74.29	**53.33**	55.05	24.39
lt	lv	direct	**63.68**	**25.88**	43.46	11.99	52.43	13.54	61.05	18.87	88.85	82.82
lt	lv	triang	**61.86**	**25.94**	43.13	9.23	52.43	13.68	57.95	17.45	88.85	82.82
sl	hr	direct	**77.89**	**62.58**	49.36	29.93	51.01	32.03	72.87	55.70	91.74	85.84
sl	hr	triang	**81.32**	**67.51**	75.02	56.90	69.78	48.94	78.63	63.96	91.74	85.84

performed better than smaller direct dictionaries, despite the triangulated dictionaries being of worse quality. Language pairs with similar languages performed better than when the training language was English. The exception is the evaluation on Latvian, where the model trained on English performed better than the model trained on Lithuanian. For evaluation on Lithuanian, both models, trained on English and Latvian, outperform the Lithuanian model. This indicates a poorly trained Lithuanian model, which explains the aforementioned exception in the evaluation of Latvian. The Lithuanian model's low performance can be partially explained by the small size of the Lithuanian treebank dataset, as seen in Table 4.

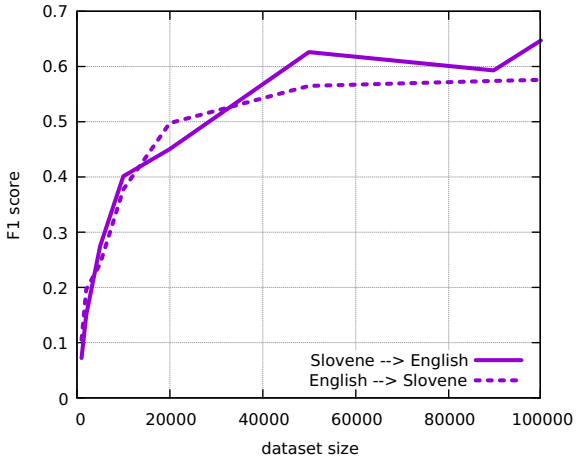
The MUSE method is stable on the DP task, which is not the case on the NER task. MUSE performs on par with Vecmap on a few language pairs. Still, its results lie somewhere between Vecmap and ELMoGAN on average.

5.3 Dataset size importance

We tested the importance of dataset size on the English-Slovene language pair. In the contextual dataset creation, we used a large, high-quality Oxford English-

Slovene dictionary instead of wiktionary. We kept all the other resources and settings the same. We evaluated ELMoGAN-10k on NER and DP tasks using various sizes of the dataset to train contextual alignments. One of the dataset sizes is 89 800 entries, which is the same size as the dataset created with a low-quality Wiktionary dictionary. We included that size for easier comparison between both dictionaries.

Fig. 2 & Table 6: Comparison of different sizes of cross-lingual contextual datasets based on different dictionaries used for cross-lingual mapping of contextual ELMo embeddings, evaluated on the NER task. LQsize represents the size of the dataset equal to the size of low quality dictionary (89 800 entries). The mapping method used was ELMoGAN-10k.



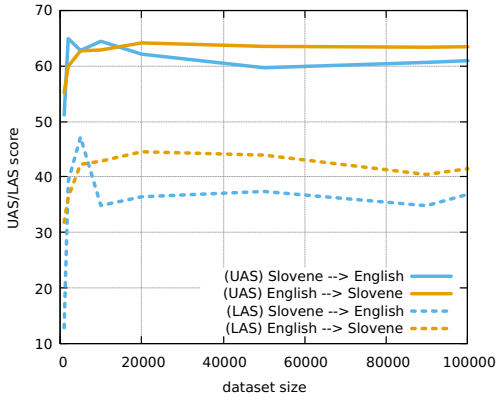
Train	Eval.	size	macro F_1
en	sl	1k	0.073
en	sl	2k	0.151
en	sl	5k	0.275
en	sl	10k	0.401
en	sl	20k	0.450
en	sl	50k	0.625
en	sl	100k	**0.646**
en	sl	LQsize	0.592
sl	en	1k	0.106
sl	en	2k	0.197
sl	en	5k	0.243
sl	en	10k	0.377
sl	en	20k	0.497
sl	en	50k	0.564
sl	en	100k	**0.575**
sl	en	LQsize	0.573

The results on NER task are shown in Figure 2 and Table 6. When we increase the size of the dataset, the performance on the NER task improves. The dataset size matters, and we presume that the performance would further increase with an even larger dataset. Surprisingly, the results achieved with the dataset of size 89 800 are slightly worse than the results achieved with the dataset of the same size, created with a low-quality dictionary (see Table 3). Using the Oxford English-Slovene dictionary, we achieved F_1 score 0.646 when trained on English and evaluated on Slovene. Using Wiktionary bilingual dictionary, we achieve F_1 score 0.664 on the same language pair with the same alignment method.

The results in the DP task show different behavior. At first, the performance quickly increases with larger datasets, and then it slowly starts to drop (see Figure 3 and Table 7). The best results are achieved with the dataset of size 20 000 when mapping from English to Slovene. When mapping from Slovene to

English, datasets of size only 2000 (based on UAS) and 5000 (based on LAS) achieve the best results.

Fig. 3 & Table 7: Comparison of different sizes of cross-lingual contextual datasets based on different dictionaries used for cross-lingual mapping of contextual ELMo embeddings, evaluated on the DP task. LQsize represents the size of the dataset based on the low quality dictionary (89 800 entries). We used the ELMoGAN-10k mapping method.



Train	Eval.	size	UAS	LAS
en	sl	1k	51.13	12.94
en	sl	2k	**64.85**	39.24
en	sl	5k	62.72	**47.17**
en	sl	10k	64.37	34.89
en	sl	20k	62.06	36.45
en	sl	50k	59.63	37.41
en	sl	100k	60.89	36.89
en	sl	LQsize	60.58	34.84
sl	en	1k	55.14	31.87
sl	en	2k	59.89	36.41
sl	en	5k	62.66	42.27
sl	en	10k	62.81	42.85
sl	en	20k	**64.09**	**44.53**
sl	en	50k	63.45	43.93
sl	en	100k	63.39	41.50
sl	en	LQsize	63.30	40.45

The better performance of wiktionary bilingual dictionary over high-quality Oxford dictionary, when datasets are of the same size, is observed in DP task as well. On Slovene to English mapping, the dataset from Oxford dictionary scores 60.58% UAS and 34.84% LAS. Wiktionary-based dataset scores 64.98% UAS and 44.86% LAS.

The results on dataset size and results from Sections 5.1 and 5.2 lead us to the conclusion that the quality of the dictionary used does not play a large role. The more important parameter is the size of the dictionary. On the NER task, larger dictionary sizes always improve results. The DP task results remain inconclusive, as the larger dictionary created with triangulation outperformed the smaller direct dictionary for similar language pairs on the DP task.

5.4 Vecmap optimizations

In computing cross-lingual alignments of two languages, the Vecmap method changes both embedding spaces. This means that we have to train a separate embedding for each language pair. In our case, we had to train eight different English models on English data for each downstream task, one for each pair of

languages, when using Vecmap for alignments. The reason is that the English vectors change during alignment as well, and we have to apply that change at the time of training cross-lingual alignment. This considerably slows down the training and evaluating procedure. We tested several approaches to avoid retraining separate models, but none was successful. The detailed description of our experiments is contained in Appendix B.

6 Conclusion

We present ELMoGAN, a novel cross-lingual mapping approach for contextual ELMO embeddings. The approach does not assume isomorphic embedding spaces and uses GANs to compute the alignments. To construct the mappings, we had to build contextual embeddings datasets for eleven language pairs. We constructed a matching set of contextual word embeddings for each language pair and each ELMo layer from parallel corpora and bilingual dictionaries. We used these new datasets to train the mappings with the proposed nonlinear mapping method ELMoGAN.

ELMoGAN is sensitive to the values of training parameters, mostly the learning rate and the number of iterations, but may bring superior performance compared to isomorphic mappings, especially for aligning more distant language pairs. To find a set of well-performing hyperparameters, this method has to be carefully fine-tuned for each task. ELMoGAN method outperformed linear mappings on the NER task but performed worse on the DP task. As this approach is not sufficiently mature, there are still open questions on the methodology for choosing the right number of iterations for each task; the dictionary induction task we currently use internally works well for the NER task but seems inappropriate for the dependency parsing task where greater emphasis is on syntactic properties of the language (and not so much on the words as in the NER task).

In further work, we intend to work on a robust method to find hyperparameters. We intend to test several more GAN architectures to find a more robust mapping. Another issue worth investigating is multiple-word terms, which are not included in current contextual mapping datasets but could be very useful in tasks requiring their joint recognition.

Acknowledgments

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 reflect only the authors' view and the EU Commission is not responsible for any use that may be made of the information it contains.

Bibliography

[1] Judit Acs. Pivot-based multilingual dictionary building using wiktionary. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation LREC*, 2014.

[2] Željko Agić and Nikola Ljubešić. Universal dependencies for Croatian (that work for Serbian, too). In *The 5th Workshop on Balto-Slavic Natural Language Processing*, pages 1–8, 2015.

[3] Hanan Aldarmaki and Mona Diab. Context-aware cross-lingual mapping. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3906–3911, 2019.

[4] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255, 2013.

[5] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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

[7] Agne Bielinskiene, Loic Boizou, and Jolanta Kovalevskaite. Lithuanian dependency treebank. In *Human Language Technologies–The Baltic Perspective: Proceedings of the Seventh International Conference Baltic HLT 2016*, volume 289, page 107, 2016.

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[9] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *Proceedings of International Conference on Learning Representation (ICLR)*, 2018.

[10] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *Proceedings of International Conference on Learning Representation ICLR*, 2018.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, 2019.

[12] Kaja Dobrovoljc, Tomaž Erjavec, and Simon Krek. The universal dependencies treebank for Slovenian. In *Proceeding of the 6th Workshop on Balto-Slavic Natural Language Processing (BSNLP 2017)*, 2017.

[13] Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *Proceedings of 5th International Conference on Learning Representations, ICLR 2017*, 2017.

[14] Chris Dyer, Victor Chahuneau, and Noah A Smith. A simple, fast, and effective reparameterization of IBM Model 2. In *Proceedings of NAACL-HLT*, pages 644–648, 2013.

[15] Zuohui Fu, Yikun Xian, Shijie Geng, Yingqiang Ge, Yuting Wang, Xin Dong, Guang Wang, and Gerard de Melo. ABSent: Cross-lingual sentence representation mapping with bidirectional GANs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7756–7763, 2020.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[17] K. Haverinen, J. Nyblom, T. Viljanen, V. Laippala, S. Kohonen, A. Missilä, S. Ojala, T. Salakoski, and F. Ginter. Building the essential resources for Finnish: the Turku dependency treebank. *LREC*, 2013.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[19] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., 2009.

[20] Simon Krek, Kaja Dobrovoljc, Tomaž Erjavec, Sara Može, Nina Ledinek, Nanika Holz, Katja Zupan, Polona Gantar, Taja Kuzman, Jaka Čibej, Špela Arhar Holdt, Teja Kavčič, Iza Škrjanec, Dafne Marko, Lucija Jezeršek, and Anja Zajc. Training corpus ssj500k 2.2, 2019. Slovenian language resource repository CLARIN.SI.

[21] Sven Laur. Nimeüksuste korpus. Center of Estonian Language Resources, 2013.

[22] Pierre Lison and Jörg Tiedemann. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation LREC*, 2016.

[23] Nikola Ljubešić, Filip Klubička, Željko Agić, and Ivo-Pavao Jazbec. New inflectional lexicons and training corpora for improved morphosyntactic annotation of Croatian and Serbian. In *Proceedings of the LREC 2016*, 2016.

[24] Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Deep multilingual correlation for improved word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 250–256, 2015.

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

[26] Kadri Muischnek, Kaili Müürisep, and Tiina Puolakainen. Estonian Dependency Treebank: from Constraint Grammar tagset to Universal Dependencies. In *Proceedings of LREC 2016*, 2016.

[27] Ndapandula Nakashole and Raphael Flauger. Characterizing departures from linearity in word translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 221–227, 2018.

[28] Joakim Nivre and Beáta Bandmann Megyesi. Bootstrapping a Swedish treebank using cross-corpus harmonization and annotation projection. In *Proceedings of Treebanks and Linguistic Theories*, 2007.

[29] Joakim Nivre, Mitchell Abrams, and Željko Agić. Universal Dependencies 2.6, 2020. URL `http://hdl.handle.net/11234/1-2988`.

[30] Aitor Ormazabal, Mikel Artetxe, Gorka Labaka, Aitor Soroa, and Eneko Agirre. Analyzing the limitations of cross-lingual word embedding mappings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics ACL*, pages 4990–4995, 2019.

[31] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 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)*, pages 2227–2237, 2018.

[32] Lauma Pretkalniņa, Laura Rituma, and Baiba Saulīte. Deriving enhanced universal dependencies from a hybrid dependency-constituency treebank. In *International Conference on Text, Speech, and Dialogue*, pages 95–105, 2018.

[33] Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. Universal dependencies for Finnish. In *Proceedings of NoDaLiDa 2015*, 2015.

[34] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[35] Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. A Finnish news corpus for named entity recognition. *Lang Resources & Evaluation*, 54(1):247–272, 2020.

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

[37] Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. A gold standard dependency corpus for English. In *Proceedings of LREC-2014*, 2014.

[38] Tadej Škvorc, Polona Gantar, and Marko Robnik-Šikonja. MICE: Mining idioms with contextual embeddings. arXiv preprint 2008.05759, 2020.

[39] Anders Søgaard, Sebastian Ruder, and Ivan Vulić. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th An-*

nual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 778–788, 2018.

[40] Anders Søgaard, Ivan Vulić, Sebastian Ruder, and Manaal Faruqui. *Cross-Lingual Word Embeddings*. Morgan & Claypool Publishers, 2019.

[41] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147, 2003.

[42] Matej Ulčar and Marko Robnik-Šikonja. High quality ELMo embeddings for seven less-resourced languages. In *Proceedings of the 12th Language Resources and Evaluation Conference, LREC 2020*, pages 4733–4740, 2020.

[43] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355, 2018.

[44] Jiawei Zhao and Andrew Gilman. Non-linearity in mapping based cross-lingual word embeddings. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3583–3589, 2020.

A Tuning the number of iterations of ELMoGAN-O

ELMoGAN mapping models have been trained for a different number of iterations for each language pair and each ELMo layer. We have trained all models for the numbers of iterations set between 6000 and 50 000. We evaluated each model on the dictionary induction task on the evaluation part of our contextual mapping dataset, presented in Section 3. We have used the average score of precision@1, precision@5, and precision@10 for both directions of our bidirectional mapping model (i.e. from the first to the second language and reverse). The number of iterations that produced the best result on the evaluation set was selected as the optimal and was used in the model called ELMoGAN-O in other evaluations. The selected numbers of iterations are presented in Table 8.

We opted not to check more than 50 000 iterations because the precision on the evaluation task rises quite quickly and then saturates or drops. For example, on the English-Finnish pair, the selected iterations were 40 000 for layer 1 and 50 000 for layers 2 and 3. Still, these numbers do not fully reflect the optimal behavior for all languages and dictionaries. The precision scores for different iterations for this language pair are shown in Figure 4.

B Vecmap speed-up experiments

As explained in Section 5.4, the Vecmap method changes both languages' embedding spaces when computing the cross-lingual alignments. This means that we have to train a separate embedding for each language pair; in our case, we had to train eight different English models, one for each pair of languages. This

Table 8: The number of iterations ELMoGAN-O was trained for, for each embedding layer and language pair. The optimal number of iterations was determined on the dictionary induction task.

Language 1	Language 2	Dictionary	Layer 1	Layer 2	Layer 3
English	Croatian	direct	15000	50000	30000
English	Estonian	direct	12000	50000	40000
English	Finnish	direct	40000	50000	50000
English	Latvian	direct	10000	40000	50000
English	Lithuanian	direct	30000	50000	40000
English	Slovenian	direct	30000	50000	25000
English	Swedish	direct	50000	50000	50000
Croatian	Slovenian	direct	15000	40000	10000
Croatian	Slovenian	triangular	25000	50000	25000
Estonian	Finnish	direct	30000	25000	15000
Estonian	Finnish	triangular	30000	50000	20000
Latvian	Lithuanian	direct	25000	40000	30000
Latvian	Lithuanian	triangular	50000	30000	25000

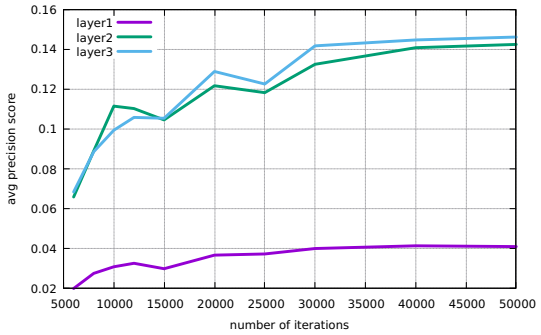


Fig. 4: The average precision score on dictionary induction task for English-Finnish alignment at different numbers of iterations of alignment algorithm.

considerably slows down the training and evaluating procedure. We tested six different sets of options for the Vecmap method to avoid retraining separate models. In this experiment, the training language was always English, and we used the DP task. By default, Vecmap first normalizes both vector sets of a language pair. Then it calculates the mapping matrix, which maps vectors from one language to the other language; in our experiment, from each language to English. Finally, it re-weighs both sets of vectors. In the results below, we denote this approach as "ELMoVM". It is identical to how we used the Vecmap method elsewhere in this paper. We tested five alternative approaches on the DP task; all of them were unsuccessful. This extra step of mapping both source and target languages seems to be unavoidable. The results for all the approaches are shown in Table 9.

Table 9: Various options used with Vecmap method on DP task. Train language is always English.

Eval.	ELMoVM		et		orth		nonorm		evalnorm		def	
lang.	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
hr	**73.96**	**60.53**	26.79	1.83	13.22	1.14	17.03	2.29	25.67	6.10	16.54	0.72
et	**62.08**	**40.62**	62.08	40.62	11.97	1.21	9.38	0.76	20.05	1.62	13.82	1.53
fi	**64.40**	**45.32**	11.38	0.76	15.41	0.49	18.11	0.62	24.50	1.94	18.53	0.83
lv	**77.84**	**65.97**	25.32	2.21	12.82	1.26	12.88	0.63	29.10	7.89	20.39	1.96
lt	**67.92**	**39.62**	9.43	0.00	7.55	0.00	7.55	0.00	15.09	0.00	11.32	1.89
sl	**79.01**	**59.84**	28.92	2.53	13.72	0.87	12.06	0.48	25.22	6.45	14.33	0.83
sv	**82.08**	**72.74**	26.23	5.23	13.50	1.46	11.27	0.81	26.45	11.92	15.22	1.41

Table 10: Options used (y) or not used (n) for different alternative methods of Vecmap mapping.

Method	ELMoVM	et	orth	nonorm	evalnorm	def
Train lang. mapped	y	y	n	n	n	n
Normalization at train time	y	y	n	n	n	n
Eval lang. mapped	y	y	y	y	y	y
Normalization at eval time	y	y	n	n	y	y
Normalization used for mapping calc.	y	y	y	n	n	y

The options for all the approaches are summarized in Table 10. The approach "et" is identical to ELMoVM, except that we used the English model trained for Estonian-English pair for all language pairs. The following four approaches do not alter English vectors in any way. Approach "orth" removes the normalization performed during the evaluation, but the normalization was still used for both languages to calculate the mapping matrix. Method "nonorm" is identical to "orth", except that we removed the normalization also during the mapping matrix calculation. Method "evalnorm" adds the normalization during the evaluation but does not use it during the mapping matrix calculation. Finally, the approach "def" uses the normalization both during the evaluation and mapping matrix calculation.

Appendix B: Cross-lingual Transfer of Sentiment Predictors

Cross-lingual Transfer of Sentiment Classifiers

Marko Robnik-Šikonja[1], Kristjan Reba[1], Igor Mozetič[2]

[1]University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, SI-1000 Ljubljana, Slovenia
marko.robnik@fri.uni-lj.si kr3377@student.uni-lj.si

[2]Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana, Slovenia
igor.mozetic@ijs.si

Abstract

Word embeddings represent words in a numeric space so that semantic relations between words are encoded as distances and directions in the vector space. Cross-lingual word embeddings map one language's vector space to the vector space of another language or vector spaces of multiple languages to the joint vector space where similar words are aligned. Cross-lingual embeddings can be used to transfer machine learning models between languages, thereby compensating for insufficient data in less-resourced languages. We use cross-lingual word embeddings to transfer machine learning prediction models for Twitter sentiment between 13 languages. We focus on two transfer mechanisms that recently show superior transfer performance. The first mechanism uses the transfer of trained models whose input is the joint numerical space for many languages as implemented in the LASER library. The second mechanism uses large pretrained multilingual BERT language models. Our experiments show that the transfer of models between similar languages is sensible, even with no target language data. The performance of cross-lingual models obtained with the multilingual BERT and LASER library is comparable, and the differences are language-dependent. The transfer with CroSloEngual BERT, pretrained on only three languages, is superior on these and some closely related languages.

1. INTRODUCTION

Word embeddings are representations of words in numerical form, as vectors of typically several hundred dimensions. The vectors are used as input to machine learning models; for complex language processing tasks, these generally are deep neural networks. The embedding vectors are obtained from specialised neural network-based embedding algorithms, e.g., word2vec (Mikolov et al., 2013) or fastText (Bojanowski et al., 2017). Modern word embedding spaces exhibit similar structures across languages, even when considering distant language pairs like English and Vietnamese (Mikolov et al., 2013). This means that embeddings independently produced from monolingual text resources can be aligned, resulting in a common cross-lingual representation, called cross-lingual embeddings, which allows for fast and effective integration of information in different languages.

There exist several approaches to cross-lingual embeddings. The first group of approaches uses monolingual embeddings with the optional help from a bilingual dictionary to align the pairs of embeddings (Artetxe et al., 2018a). The second group of approaches uses bilingually aligned (comparable or even parallel) corpora to construct joint embeddings (Artetxe and Schwenk, 2019). This approach is implemented in the LASER library[1] and is available for 93 languages. The third type of approaches is based on large pretrained multilingual masked language models such as BERT (Devlin et al., 2019). In this work, we focus on the second and third group of approaches. In particular, from the third group, we apply two variants of BERT models, the original multilingual BERT model (mBERT), trained on 104 languages, and trilingual CroSloEngual BERT (Ulčar and Robnik-Šikonja, 2020) trained on Croatian, Slovene, and English (CSE BERT).

Sentiment annotation is a costly and lengthy operation, with a relatively low inter-annotator agreement (Mozetič et al., 2016). Large annotated sentiment datasets are, therefore, rare, especially for low-resourced languages. The transfer of already trained models or datasets from other languages would be useful. It would increase the ability to study sentiment-related phenomena for many more languages than possible today.

Our study aims to analyse the abilities of modern cross-lingual approaches for the transfer of trained models between languages.

We study two cross-lingual transfer technologies, using a joint vector space computed from parallel corpora with the LASER library and multilingual BERT models. The advantage of our study is sizeable comparable classification datasets in 13 different languages, which gives credibility and general validity to our findings. Further, due to the datasets' size, we can reliably test different transfer modes: direct transfer between languages (called a zero-shot transfer) and transfer with enough fine-tuning data in the target language. In the experiments, we study two cross-lingual transfer modes based on projections of sentences into a joint vector space. The first mode transfers trained models from source to target languages. The model is trained on the source language(s) and used for classification in the target language(s). This model transfer is possible because texts in all processed languages are embedded into the common vector space. The second mode expands the training set with instances from other languages, and then all instances are mapped into the common vector space during neural network training. Besides the cross-lingual transfer, we analyse the quality of representations for the Twitter sentiment classification and compare the common vector space for several languages constructed by the LASER library, multilingual BERT models, and the traditional bag-of-words approach. The results show a relatively low decrease in predictive performance when transferring trained

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

sentiment prediction models between similar languages and superior performance of multilingual BERT models covering only three languages.

The paper is divided into four more sections. In Section 2, we present background on different types of cross-lingual embeddings: alignment of monolingual embeddings, building a fixed common vector space for several languages, and large pretrained multilingual contextual models. We also discuss related work on Twitter sentiment analysis and cross-lingual transfer of the classification model. In Section 3, we present a large collection of tweets from 13 languages used in our empirical evaluation, the implementation details of our deep neural network prediction models, and the evaluation metrics used. Section 4 contains four series of experiments. We first analyse the transfer of trained models between languages from the same language group and from a different language group, followed by expanding datasets with instances from other languages. We end the experimental part by evaluating representation spaces and comparing the common vector space with the multilingual BERT models. In Section 5, we summarise the results and present ideas for further work.

2. BACKGROUND AND RELATED WORK

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.

Cross-lingual approaches can be sorted into three groups, described in the following three subsections. The first group of methods uses monolingual embeddings with (an optional) help from bilingual dictionaries to align the embeddings. The second group of approaches uses bilingually aligned (comparable or even parallel) corpora for joint construction of embeddings in all handled languages. The third type of approach is based on large pretrained multilingual masked language models such as BERT (Devlin et al., 2019). The multilingual BERT is typically used as a starting model, which is fine-tuned for a particular task without explicitly extracting embedding vectors.

In Section 2.1, we first present background information on the alignment of individual monolingual embeddings. We describe the projections of many languages into a joint vector space in Section 2.2, and in Section 2.3, we present variants of multilingual BERT models. In Section 2.4, we describe related work on Twitter sentiment classification. Finally, in Section 2.5, we outline the cross-lingual transfer of classification models.

2.1. Alignment of monolingual embeddings

Cross-lingual alignment methods take precomputed word embeddings for each language and align them with the 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 joint 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., 2018a), named *vecmap*[2], can align monolingual embeddings using a supervised, semi-supervised, or unsupervised approach.

[2] https://github.com/artetxem/vecmap

The supervised approach requires the use of a bilingual dictionary, which is used to match embeddings of equivalent words. The embeddings are aligned using the Moore-Penrose pseudo-inverse, which minimises the sum of squared Euclidean distances. The algorithm always converges but can be caught in a local maximum. Several methods (e.g., stochastic dictionary introduction or frequency-based vocabulary cut-off) are used to help the algorithm climb out of local maxima. 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 low but sufficient quality for later processing. After the initial dictionary (either by seeding dictionary or using similarity matrices) is built, an iterative algorithm is applied. The algorithm first computes optimal mapping using the pseudo-inverse approach for the given initial dictionary. The optimal dictionary for the given embeddings is then computed, and the procedure is repeated with the new dictionary.

When constructing mappings between embedding spaces, a bilingual dictionary can help as its entries can be used as anchors for the alignment map for supervised and semi-supervised approaches. However, lately, researchers have proposed methods that do not require a bilingual dictionary but rely on the adversarial approach (Conneau et al., 2018) or use the words' frequencies (Artetxe et al., 2018b) to find a required transformation. These are called unsupervised approaches.

2.2. Projecting into a joint vector space

To construct a common vector space for all the processed languages, one requires 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 the 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 (Language-Agnostic SEntence Representations) is a Facebook research project focusing on joint sentence representation for many languages (Artetxe and Schwenk, 2019). Similarly to machine translation architectures, it uses an encoder-decoder architecture. The encoder is trained on a large parallel corpus, 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 many languages; currently, the encoder supports 93 different languages. Using LASER, one can train a classifier on data from just one language and use it on any language supported by LASER. A vector representation in the joint embedding space can be transformed back into a sentence using a decoder for the specific language.

2.3. Multilingual BERT and CroSloEngual BERT

BERT (Bidirectional Encoder Representations from Transformers) embedding (Devlin et al., 2019) generalises the idea of a language model (LM) to masked language models, inspired by the cloze test, which tests understanding of a text by removing a few 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 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 sub-word units. The input is constructed by

summing the embeddings of corresponding tokens, segments, and positions. Some widespread words are kept as single tokens; others are split into sub-words (e.g., frequent stems, prefixes, suffixes—if needed down to single letter tokens). The original BERT project offers pre-trained English, Chinese, and multilingual model. The latter, called mBERT, is trained on 104 languages simultaneously.

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 maximise the log-probability of the correct labels.

Recently, a new type of multilingual BERT models emerged that reduce the number of languages in multilingual models. For example, CSE BERT (Ulčar & Robnik-Šikonja, 2020) uses Croatian, Slovene (two similar less-resourced languages from the same language family), and English. The main reasons for this choice are to represent each language better and keep sensible sub-word vocabulary, as shown by Virtanen et al. (2019). This model is built with the cross-lingual transfer of prediction models in mind. As CSE BERT includes English, we expect that it will enable the better transfer of existing prediction models from English to Croatian and Slovene.

2.4. Twitter sentiment classification

We present a brief overview of the related work on automated sentiment classification of Twitter posts. We summarise the published labelled sets used for training the classification models and the machine learning methods applied for training. Most of the related work is limited to English texts only.

To train a sentiment classifier, one needs a reasonably large training dataset of tweets already labelled with the sentiment. One can rely on a proxy, e.g., emoticons used in the tweets, to determine the intended sentiment; however, high-quality labelling requires the engagement of human annotators. There exist several publicly available and manually labelled Twitter datasets. They vary in the number of examples from several hundred to several thousand, but to the best of our knowledge, so far, none exceeds 20,000 entries. Saif et al. (2013) describe eight Twitter sentiment datasets and introduce a new one that contains separate sentiment labels for tweets and entities. Rosenthal et al. (2015) provide statistics for several of the 2013–2015 SemEval datasets.

There are several supervised machine learning algorithms suitable to train sentiment classifiers from sentiment labelled tweets. For example, in the SemEval-2015 competition, before the rise of deep neural networks, the most often used algorithms for the sentiment analysis on Twitter (Rosenthal et al., 2015) were support vector machines (SVM), maximum entropy, conditional random fields, and linear regression. In other cases, frequently used classifiers were naive Bayes, k-nearest neighbours, and even decision trees. Often, SVM was shown as the best performing classifier for the Twitter sentiment. However, only recently, when researchers started to apply deep learning for Twitter sentiment classification, considerable improvements in classification performance were observed (Wehrmann et al., 2017; Jianqiang et al., 2018; Naseem et al., 2020). Similarly to our approach, recent approaches use contextual embeddings such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), but in a monolingual setting.

2.5. Transfer of trained models

Cross-lingual word embeddings can be used directly as inputs in NLP models. The main idea is to train a model on data from one language and then apply it to another, relying on shared cross-lingual representation. Several tasks have been attempted in testing cross-lingual transfer and use different approaches. Søgaard et al. (2019) survey the transfer in the following tasks: document classification, dependency parsing, POS tagging,

named entity recognition (NER), super-sense tagging, semantic parsing, discourse parsing, dialogue state tracking (DST), entity linking (wikification), sentiment analysis, machine translation, natural language interference, etc. For example, Ranasinghe and Zampieri (2020) apply large pretrained models in a similar way as we but use offensive language domain and only four languages from different families (English, Spanish, Bengali, and Hindu). In sentiment analysis, which is of particular interest in this work here, Mogadala and Rettinger (2016) evaluate their embeddings on the multilingual Amazon product review dataset. In the Twitter sentiment analysis, Wehrmann et al. (2017) use LSTM networks but first learn a joint representation for four languages (English, German, Portuguese, and Spanish) with character-based convolutional neural networks.

3. DATASETS AND EXPERIMENTAL SETTINGS

This section presents the evaluation metrics, experimental data, and implementation details of the used neural prediction models.

3.1. Evaluation metrics

Following Mozetič et al. (2016), we report the $\bar{F}_1$ score and classification accuracy (*CA)*. $F_1(c)$ score for class value c is the harmonic mean of precision p and recall r for the given class c, where the precision is defined as the proportion of correctly classified instances from the instances predicted to be from the class c, and the recall is the proportion of correctly classified instances actually from the class c:

$$F_1(c) = \frac{2p_c r_c}{p_c + r_c}.$$

The F_1 score returns values from the [0,1] interval, where 1 means perfect classification, and 0 indicates that either precision or recall for class c is 0. We use an instance of the F_1 score specifically designed to evaluate the 3-class sentiment models (Kiritchenko et al., 2014). $\bar{F}_1$ is defined as the average over the positive (+) and negative (−) sentiment class:

$$\bar{F}_1 = \frac{F_1(+) + F_1(-)}{2}.$$

$\bar{F}_1$ implicitly considers the ordering of sentiment values by considering only the extreme labels, positive (+) and negative (-). The middle, neutral, is taken into account indirectly. $\bar{F}_1$=1 implies that all negative and positive tweets were correctly classified, and as a consequence, all neutrals as well. $\bar{F}_1$ = 0 indicates that all tweets were classified as neutral, and consequently, all negative and positive tweets were incorrectly classified.

$\bar{F}_1$ is not the best performance measure. First, taking the arithmetic average of the F_1 scores over different classes (called macro F_1) is methodologically misguided (Flach and Kull, 2015). It is justified only when the class distribution is approximately even, as is in our case. Second, $\bar{F}_1$ does not account for correct classification by chance. A more appropriate measure that allows for class ordering, classification by chance, and class labelling with disagreements is Krippendorff's alpha-reliability (Krippendorff, 2013). However, since $\bar{F}_1$ is commonly used in the sentiment classification community, and the results are typically well correlated with the alpha-reliability, we decided to report our experimental results in terms of $\bar{F}_1$.

The second score we report is the classification accuracy CA, defined as the ratio of correctly predicted tweets N_c to all the tweets N:

$$CA = \frac{N_c}{N}.$$

3.2. Datasets

We use a corpus of Twitter sentiment datasets (Mozetič et al., 2016), consisting of 15 languages, with over 1.6 million annotated tweets. The languages covered are Albanian, Bosnian, Bulgarian, Croatian, English, German, Hungarian, Polish, Portuguese, Russian, Serbian, Slovak, Slovene, Spanish, and Swedish. The authors studied the annotators' agreement on the labelled tweets. They discovered that the SVM classifier achieves significantly lower score for some languages (English, Russian, Slovak) than the annotators. This hints that there might be room for improvement for these languages using a better classification model or larger training set.

We cleaned the above datasets by removing the duplicated tweets, weblinks, and hashtags. Due to the low quality of sentiment annotations indicated by low self-agreement and low inter-annotator agreement, we removed Albanian and Spanish datasets. For these two languages, the self-agreement expressed with $\bar{F}_1$ score is 0.60 and 0.49, respectively; the inter-annotator agreement is 0.41 and 0.42. As defined above, $\bar{F}_1$ is the arithmetic average of F_1 scores for the positive and negative tweets, where $F_1(c)$ is the fraction of equally labelled tweets out of all the tweets with the label c.

In the paper where the datasets were introduced (Mozetič et al., 2016), Serbian, Croatian, and Bosnian tweets were merged into a single dataset. The three languages are very similar and difficult to distinguish in short Twitter posts. However, it turned out that this merge resulted in poor classification performance due to a very different quality of annotations. In particular, Serbian (71,721 tweets) was annotated by 11 annotators, where two of them accounted for over 40% of the annotations. All the inter-annotator agreement measures come from the Serbian only (1,880 tweets annotated twice by different annotators, $\bar{F}_1$ is 0.51), and there are very few tweets annotated twice by the same annotator (182 tweets only, $\bar{F}_1$ for the self-agreement is 0.46). In contrast, all the Croatian and Bosnian tweets were annotated by a single annotator, and we have reliable self-agreement estimates. There are 84,001 Croatian tweets, 13,290 annotated twice, and the self-agreement $\bar{F}_1$ is 0.83. There are 38,105 Bosnian tweets, 6,519 annotated twice, and the self-agreement $\bar{F}_1$ is 0.78. The authors concluded that the annotation quality of the Croatian and Bosnian tweets is considerably higher than that of the Serbian. If one constructs separate sentiment classifiers for each language, one observes a very different performance than reported originally. The individual classifiers are better and "well-behaved" compared to the joint Serbian/Croatian/Bosnian model. In this paper, we follow the authors' suggestion that datasets with no overlapping annotations and different annotation quality are better not merged. As a consequence, the Serbian, Croatian, and Bosnian datasets are analysed separately. The characteristics of all the 13 datasets are presented in Table 1.

	Number of tweets				Agreement ($\bar{F}_1$)	
Language	Negative	Neutral	Positive	All	Self-	Inter-
Bosnian	12,868	11,526	13,711	38,105	0.78	-
Bulgarian	15,140	31,214	20,815	67,169	0.77	0.50
Croatian	21,068	19,039	43,894	84,001	0.83	-
English	26,674	46,972	29,388	103,034	0.79	0.67
German	20,617	60,061	28,452	109,130	0.73	0.42
Hungarian	10,770	22,359	35,376	68,505	0.76	-
Polish	67,083	60,486	96,005	223,574	0.84	0.67
Portuguese	58,592	53,820	44,981	157,393	0.74	-
Russian	34,252	44,044	29,477	107,773	0.82	-
Serbian	24,860	30,700	16,161	71,721	0.46	0.51
Slovak	18,716	14,917	36,792	70,425	0.77	-
Slovene	38,975	60,679	34,281	133,935	0.73	0.54
Swedish	25,319	17,857	15,371	58,547	0.76	-

Table 1: The left-hand side reports the number of tweets from each category and the overall number of instances for individual languages. The right-hand side contains self-agreement of annotators and inter-annotator agreement for tried languages where more than one annotator was involved.

3.3. Implementation details

In our experiments, we use three different types of prediction models, BiLSTM neural networks using joint vector space embeddings constructed with the LASER library, and two variants of BERT, mBERT, and CSE BERT. The original mBERT (bert-multi-cased) is pretrained on 104 languages, has 12 transformer layers, and 110 million parameters. The CSE BERT uses the same architecture but is pretrained only on Croatian, Slovene, and English. In the construction of sentiment classification models, we fine-tune the whole network, using the batch size of 32, 2 epochs, and Adam optimiser. We also tested larger numbers of epochs and larger batch sizes in preliminary experiments, but this did not improve the performance.

The cross-lingual embeddings from the LASER library are pretrained on 93 languages, using BiLSTM networks, and are stored as 1024 dimensional embedding vectors. Our classification models contain an embedding layer, followed by a multilayer perceptron hidden layer of size 8, and an output layer with three neurons (corresponding to three output classes, negative, neutral, and positive sentiment) using the softmax. We use the ReLU activation function and Adam optimiser. The fine-tuning uses a batch size of 32 and 10 epochs.

Further technical details are available in the freely available source code.

4. EXPERIMENTS AND RESULTS

Our experimental work focuses on model transfer with cross-lingual embeddings. However, to first establish the suitability of different embedding spaces for Twitter sentiment classification, we start with their comparison in a monolingual setting in Section 4.1. We compare the three neural approaches presented in Section 3.3 (common vector space of LASER, mBERT, and CSE BERT). As a baseline, we use the classical approach using bag-of-ngram representation with the SVM classifier. In the cross-lingual experiments, we focus on the two most-successful types of model transfer, described in Sections 2.2 and 2.3: the common vector space of the LASER library and the variants of the multilingual BERT model (mBERT and CSE BERT). We conducted several cross-lingual transfer experiments: transfer of models between languages from the same (Section 4.2) and different language family (Section 4.3), as well as the expansion of training sets with varying amounts of data from other languages (Section 4.4). In the experiments, we did not systematically test all possible combinations of languages and language groups as this would require an excessive amount of computational time and would not contribute to the clarity of the paper. Instead, we arbitrarily selected a representative set of language combinations in advance. We leave a comprehensive systematic approach based on informative features (Lin et al., 2019) for further work.

4.1. Comparing embedding spaces

To establish the appropriateness of different embedding approaches for our Twitter sentiment classification task, we start with experiments in a monolingual setting. We compare embeddings into a joint vector space obtained with the LASER library with the mBERT and CSE BERT. Note that there is no transfer between

different languages in this experiment but only a test of the suitability of the representation, i.e., embeddings. To make the results comparable with previous work on this dataset, we report results obtained with 10-fold blocked cross-validation. There is no randomisation of training examples in the blocked cross-validation, and each fold is a block of consecutive tweets. It turns out that standard cross-validation with a random selection of examples yields unrealistic estimates of classifier performance and should not be used to evaluate classifiers in time-ordered data scenarios (Mozetič et al., 2018).

As a baseline, we report the results of the SVM models without neural embeddings that use Delta TF-IDF weighted bag-of-ngrams representation with substantial preprocessing of tweets (Mozetič et al., 2016). Further, the datasets for the Bosnian, Croatian, and Serbian languages were merged in (Mozetič et al., 2016) due to the similarity of these languages; therefore, we report the performance on the merged dataset for the SVM classifier. Results are presented in Table 2.

	LASER		mBERT		CSE BERT		SVM	
Language	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA
Bosnian	**0.68**	0.64	0.65	0.60	**0.68**	**0.65**	(0.61	0.56)
Bulgarian	0.53	**0.59**	**0.58**	**0.59**	0.00	0.45	0.52	0.54
Croatian	0.72	0.68	0.64	0.66	**0.76**	**0.71**	(0.61	0.56)
English	0.62	0.65	**0.68**	**0.68**	0.67	0.66	0.63	0.64
German	0.52	0.64	**0.66**	**0.66**	0.31	0.59	0.54	0.61
Hungarian	0.63	0.67	**0.65**	**0.69**	0.57	0.65	0.64	0.67
Polish	**0.70**	0.66	**0.70**	**0.70**	0.56	0.57	0.68	0.63
Portuguese	0.48	0.47	0.50	0.49	0.12	0.22	**0.55**	**0.51**
Russian	**0.70**	**0.70**	0.64	0.64	0.07	0.43	0.61	0.60
Serbian	0.50	0.54	0.50	0.52	0.30	0.50	(**0.61**	**0.56**)
Slovak	**0.72**	**0.72**	0.67	0.66	0.69	0.71	0.68	0.68
Slovene	0.57	0.58	0.58	0.58	**0.60**	**0.61**	0.55	0.54
Swedish	**0.67**	0.64	**0.67**	**0.65**	0.54	0.56	0.66	0.62
#Best	5	3	6	6	3	3	2	2

Table 2: Comparison of different representations: supervised mapping into a joint vector space with the LASER library, mBERT, CSE BERT, and bag-of-ngrams with the SVM classifier. The best score for each language and metric is in bold. In the last row, we count the number of best scores for each model. The SVM results for Bosnian, Croatian, and Serbian were obtained with the model trained on the merged dataset of these languages model and are therefore not directly compatible with the language-specific results for the other representations.

The SVM baseline using bag-of-ngrams representation mostly achieves lower predictive performance than the two neural embedding approaches. We speculate that the main reason is more information about the language structure contained in precomputed dense embeddings used by the neural approaches. Together with the fact that standard feature-based machine learning approaches require much more preprocessing effort, it seems that there are no good reasons why to bother with this approach in text classification; we, therefore, omit this method from further experiments. The mBERT model is the best of the tested methods, achieving the best $\bar{F}_1$ and CA scores in six languages (in bold), closely followed by the LASER approach, which achieves the best $\bar{F}_1$

score in five languages and the best CA score in three languages. The CSE BERT is specialised for only three languages, and it achieves the best scores in languages where it is trained (except in English, where it is close behind the mBERT), and in Bosnian, which is similar to Croatian. Overall, it seems that large pretrained transformer models (mBERT and CSE BERT) are dominating in the Twitter sentiment prediction. The downside of these models is that their training, fine-tuning, and execution require more computational time than precomputed fixed embeddings. Nevertheless, with progress in optimisation techniques for neural network learning and advent of computationally more efficient BERT variants, e.g., (You et al., 2020), this obstacle might disappear in the future.

4.2. Transfer to the same language family

The transfer of prediction models between similar languages from the same language family is the most likely to be successful. We test several combinations of source and target languages from Slavic and Germanic language families. We report the results in Table 3.

In each experiment, we use the entire dataset(s) of the source language as the training set and the whole dataset of the target language as the testing set, i.e., we do a zero-shot transfer. We compare the results with the LASER embeddings with BiLSTM network using training and testing set from the target language, where 70% of the dataset is used for training and 30% for testing. As we use large datasets, the latter results can be taken as an upper bound of what cross-lingual transfer models could achieve in ideal conditions.

The results from Table 3 (bottom line) show that there is a gap in the performance of transfer learning models and native models. On average, the gap in $\bar{F}_1$ is 5% for the LASER approach, 6% for mBERT, and 8% for CSE BERT. For CA, the average gap is 7% for both LASER and mBERT and 8% for CSE BERT. However, there are significant differences between languages, and we advise to test both variants for a specific language, as the models are highly competitive. The CSE BERT is slightly less successful measured with the average performance gap over all languages as the gap is 8% in both $\bar{F}_1$ and CA. However, if we take only the three languages used in training of CSE BERT (Croatian, Slovene, and English) as shown in Table 4, conclusions are entirely different. The average performance gap is 0% in $\bar{F}_1$ and 1% in the classification accuracy, meaning that we get almost a perfect cross-lingual transfer for these languages on the Twitter sentiment prediction task.

We also tried more than one input language at once, for example, German and Swedish as source languages and English as the target language, as shown in Table 3. The success of the tested combinations is mixed: for some models and some languages, we slightly improve the scores, while for others, we slightly decrease them. We hypothesise that our datasets for individual languages are large enough so that adding additional training data does not help.

		LASER		mBERT		CSE BERT		Both target	
Source	Target	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA
German	English	0.55	0.59	**0.63**	**0.64**	0.42	0.42	0.62	0.65
English	German	0.55	0.60	**0.66**	**0.70**	0.50	0.58	0.53	0.65
Polish	Russian	**0.64**	**0.59**	0.57	0.57	0.50	0.40	0.70	0.70
Polish	Slovak	**0.63**	0.59	0.58	0.59	**0.63**	**0.65**	0.72	0.72
German	Swedish	0.58	0.57	**0.59**	**0.59**	0.58	0.56	0.67	0.65
German Swedish	English	**0.58**	**0.60**	0.55	0.56	0.41	0.42	0.62	0.65
Slovene Serbian	Russian	0.53	0.55	0.57	**0.57**	**0.58**	0.48	0.70	0.70

Slovene Serbian	Slovak	**0.59**	0.52	0.57	0.59	0.48	**0.60**	0.72	0.72
Serbian	Slovene	0.54	**0.57**	0.54	0.54	**0.56**	0.55	0.60	0.60
Serbian	Croatian	**0.67**	0.64	0.65	0.62	0.65	**0.70**	0.73	0.68
Serbian	Bosnian	**0.65**	0.61	0.61	0.60	0.59	**0.62**	0.67	0.64
Polish	Slovene	0.51	0.48	**0.55**	**0.54**	0.50	0.53	0.60	0.60
Slovak	Slovene	0.52	0.51	0.54	0.54	**0.58**	**0.58**	0.60	0.60
Croatian	Slovene	0.53	0.53	0.53	0.54	**0.61**	**0.60**	0.60	0.60
Croatian	Serbian	**0.54**	**0.52**	0.52	0.51	0.52	0.49	0.48	0.54
Croatian	Bosnian	0.66	0.61	0.57	0.56	**0.67**	**0.62**	0.67	0.64
Slovene	Croatian	0.70	0.65	0.64	0.63	**0.73**	**0.69**	0.73	0.68
Slovene	Serbian	**0.52**	**0.55**	0.46	0.49	0.47	0.50	0.48	0.54
Slovene	Bosnian	**0.66**	0.61	0.58	0.56	**0.66**	**0.62**	0.67	0.64
Average performance gap		0.05	0.07	0.06	0.07	0.08	0.08		

Table 3: The transfer of trained models between languages from the same language family using LASER common vector space, mBERT, and CSE BERT. We compare the results with both training and testing set from the target language using the LASER approach (the right-most two columns).

		LASER		mBERT		CSE BERT		Both target	
Source	Target	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA
Croatian	Slovene	0.53	0.53	0.53	0.54	**0.61**	**0.60**	0.60	0.60
Croatian	English	**0.63**	0.63	**0.63**	**0.66**	0.62	0.64	0.62	0.65
English	Slovene	0.54	0.57	0.50	0.53	**0.59**	**0.57**	0.60	0.60
English	Croatian	0.62	**0.67**	0.67	0.63	**0.73**	**0.67**	0.73	0.68
Slovene	English	0.63	0.64	**0.65**	**0.67**	0.63	0.64	0.62	0.65
Slovene	Croatian	0.70	0.65	0.64	0.63	**0.73**	**0.69**	0.73	0.68
Croatian English	Slovene	0.54	0.54	0.53	0.54	**0.60**	**0.58**	0.60	0.60
Croatian Slovene	English	0.62	0.61	**0.65**	**0.67**	0.63	0.65	0.62	0.65
English Slovene	Croatian	0.64	0.68	0.63	0.63	**0.68**	**0.70**	0.73	0.68
Average performance gap		0.04	0.03	0.04	0.03	0.00	0.01		

Table 4: The transfer of sentiment models between all combinations of languages on which CSE BERT was trained (Croatian, Slovene, and English).

4.3. Transfer to a different language family

The transfer of prediction models between languages from different language families is less likely to be successful. Nevertheless, to observe the difference, we test several combinations of source and target languages from different language families (one from Slavic, the other from Germanic, and vice-versa). We compare the LASER approach with mBERT models; the CSE BERT is not constructed for these circumstances, and we skip it in this section. We report the results in Table 5.

The results show that with the LASER approach, there is an average decrease of performance for transfer learning models of 11% (both $\bar{F}_1$ and CA), and for mBERT, the gap is 9%. This gap is significant and makes the resulting transferred models less useful in the target languages, though there are considerable differences between the languages. Another observation is that the differences between target languages are significant.

		LASER		mBERT		Both target	
Source	Target	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA
Russian	English	**0.52**	0.56	**0.52**	**0.57**	0.62	0.65
English	Russian	**0.57**	**0.58**	0.55	0.57	0.70	0.70
English	Slovak	0.46	0.44	**0.57**	**0.58**	0.72	0.72
Polish, Slovene	English	0.58	0.57	**0.60**	**0.60**	0.62	0.65
German, Swedish	Russian	0.61	**0.61**	**0.62**	0.59	0.70	0.70
English, German	Slovak	0.50	0.47	**0.56**	**0.54**	0.72	0.72
German	Slovene	**0.54**	**0.56**	0.53	0.54	0.60	0.60
English	Slovene	**0.54**	**0.57**	0.50	0.53	0.60	0.60
Swedish	Slovene	**0.54**	**0.56**	0.52	0.54	0.60	0.60
Hungarian	Slovene	0.52	0.52	**0.53**	**0.54**	0.60	0.60
Portuguese	Slovene	0.51	0.49	**0.54**	**0.54**	0.60	0.60
Average performance gap		0.11	0.11	0.09	0.09		

Table 5: The transfer of trained models between languages from different language families using LASER common vector space and the mBERT. We compare the results with both training and testing set from the target language using the LASER approach (the right-most two columns).

4.4. Increasing datasets with several languages

Another type of cross-lingual transfer is possible if we increase the training sets with instances from several related and unrelated languages. We conduct two sets of experiments in this setting. In the first setting, reported in Table 6, we constructed the training set in each experiment with instances from several languages and 70% of the target language dataset. The remaining 30% of target language instances are used as the testing set. In the second setting, reported in Table 7, we merge *all* other languages and 70% of the target language into a joint training set. We compare the LASER approach, mBERT, and also CSE BERT, as Slovene and Croatian are involved in some combinations.

Table 6 shows a gap between learning models using the expanded datasets and models with only target language data. The decrease is more extensive for both BERT models (on average around 10%) than for the LASER approach (the decrease is on average 3% for $\bar{F}_1$ and 5% for CA). These results indicate that the tested expansion of datasets was unsuccessful, i.e. the provided amount of training instances in the target language was already sufficient for successful learning. The additional instances from other languages in the transformed space are likely to be of lower quality than the native instances and therefore decrease the performance.

		LASER		mBERT		CSEBERT		Target only	
Source	Target	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA
English, Croatian, Slovene	Slovene	0.58	0.53	0.46	0.45	**0.60**	**0.58**	0.60	0.60

English, Croatian, Serbian, Slovak	Slovak	**0.67**	**0.65**	0.57	0.54	0.27	0.37	0.72	0.72
Hungarian,Slovak,English,Croatian,Russian	Russian	**0.67**	**0.65**	0.61	0.59	0.63	0.61	0.70	0.70
Russian, Swedish, English	English	0.60	0.61	**0.62**	0.60	0.59	**0.62**	0.62	0.65
Croatian, Serbian, Bosnian, Slovene	Slovene	0.54	**0.58**	0.44	0.45	**0.57**	0.56	0.60	0.60
English, Swedish, German	German	0.55	0.60	**0.60**	**0.64**	0.47	0.58	0.53	0.65
Average performance gap		0.03	0.05	0.08	0.11	0.11	0.10		

Table 6: The expansion of training sets with instances from several languages. We compare the LASER approach, mBERT, and CSE BERT. As the upper bound, we give results of the LASER approach trained on only the target language.

The results in Table 7, where we test the expansion of the training set (consisting of 70% of the dataset in the target language) with all other languages, show that using many languages and significant enlargement of datasets is also not successful. The two improvements in the LASER approach over using only target language are limited to a single metric (F_1in case of Bulgarian and Serbian), which indicates that true positives are favoured at the expense of true negatives. For all other languages, the tried expansions of training sets are unsuccessful for the LASER approach, and the difference to native models is on average 3.5% for the $\bar{F}_1$ score and 6% for CA. The mBERT models are in almost all cases more successful in this massive transfer than LASER models, and they sometimes marginally beat the reference mBERT approach trained only on the target language.

	LASER				mBERT			
	All & Target		Only Target		All &Target		Only Target	
Target	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA	$\bar{F}_1$	CA
Bosnian	0.64	0.59	0.67	0.64	0.63	0.60	0.65	0.60
Bulgarian	**0.54**	0.56	0.50	0.59	**0.60**	**0.60**	0.58	0.59
Croatian	0.63	0.57	0.73	0.68	**0.65**	0.63	0.64	0.66
English	0.58	0.60	0.62	0.65	0.64	**0.69**	0.68	0.68
German	0.52	0.59	0.53	0.65	0.61	0.66	0.66	0.66
Hungarian	0.59	0.61	0.60	0.67	0.65	0.69	0.65	0.69
Polish	0.67	0.63	0.70	0.66	**0.71**	**0.71**	0.70	0.70
Portuguese	0.44	0.39	0.52	0.51	**0.52**	**0.52**	0.50	0.49
Russian	0.66	0.64	0.70	0.70	**0.67**	**0.66**	0.64	0.64
Serbian	**0.52**	0.49	0.48	0.54	**0.53**	0.51	0.50	0.52
Slovak	0.64	0.61	0.72	0.72	0.67	0.65	0.67	0.66
Slovene	0.54	0.50	0.60	0.60	0.56	0.54	0.58	0.58
Swedish	0.63	0.59	0.67	0.65	0.67	0.64	0.67	0.65
Avg. gap	0.03	0.06			0.00	0.00		

Table 7: The expansion of training sets with instances from all other languages (+70% of the target language instances) to train the LASER approach and mBERT. We compare the results with the training on only the target language. The scores where models with the expanded training sets beat their respective reference scores are in bold.

5. CONCLUSIONS

We studied state-of-the-art approaches to the cross-lingual transfer of Twitter sentiment prediction models: mappings of words into the common vector space using the LASER library and two multilingual BERT variants: mBERT and trilingual CSE BERT. Our empirical evaluation is based on relatively large datasets of labelled tweets from 13 European languages. We first tested the success of these text representations in a monolingual setting. The results show that BERT variants are the most successful, closely followed by the LASER approach, while the classical bag-of-ngrams coupled with SVM classifier is no longer competitive with these neural approaches. In the cross-lingual experiments, the results show that there is a significant transfer potential using the models trained on similar languages; compared to training and testing on the same language, with LASER, we get on average 5% lower $\overline{F}_1$ score and with mBERT 6% lower $\overline{F}_1$ score. The transfer of models with CSE BERT is even more successful in the three languages covered by this model, where we get no performance gap compared to the LASER approach trained and tested on the target language. Using models trained on languages from different language families produces more considerable differences (on average around 10% for $\overline{F}_1$ and CA). Our attempt to expand training sets with instances from different languages was unsuccessful using either additional instances from a small group of languages or instances from all other languages. The source code of our analyses is freely available[3].

We plan to expand the BERT models with additional emotional and subjectivity information in future work on sentiment classification. Given the favourable results in cross-lingual transfer, we will expand the work to other relevant tasks.

ACKNOWLEDGEMENTS

The research was supported by the Slovene Research Agency through research core funding no. P6-0411 and P2-103, as well as project no. J6-2581. This paper is supported by European Union's Horizon 2020 Programme project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media, grant no. 825153), and Rights, Equality and Citizenship Programme project IMSyPP (Innovative Monitoring Systems and Prevention Policies of Online Hate Speech, grant no. 875263). The results of this publication reflect only the author's view, and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot crosslingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Generalising and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

[3] https://github.com/kristjanreba/cross-lingual-classificationof-tweet-sentiment

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Herve J' egou. 2018. Word' translation without parallel data. In *Proceedings of International Conference on Learning Representation ICLR 2018*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186.

Flach P, Kull M. Precision-recall-gain curves: PR analysis done right, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.) Advances in Neural Information Processing Systems, 2015, pp. 838-846.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, et al. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135.

Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep convolution neural networks for Twitter sentiment analysis. *IEEE Access*, 6, 23253-23260.

Krippendorff K. *Content Analysis, An Introduction to Its Methodology. 3rd ed*. Thousand Oaks, CA, USA: Sage Publications; 2013.

Kiritchenko S, Zhu X, Mohammad SM. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*. 2014; 50:723–762.

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

Mogadala, A., and Rettinger, A. 2016. Bilingual word embeddings from parallel and non-parallel corpora for cross-language text classification. In *Proceedings of NAACL-HLT*, pp. 692–702

Igor Mozetič, Miha Grčar, and Jasmina Smailović. 2016. Multilingual Twitter sentiment classification: The role of human annotators. *PLOS ONE*, 11(5).

Igor Mozetič, Torgo L, Cerqueira V, Smailović J (2018) How to evaluate sentiment classifiers for Twitter time-ordered data? *PLoS ONE* 13(3).

Naseem, U., Razzak, I., Musial, K., & Imran, M. (2020). Transformer based deep intelligent contextual embedding for Twitter sentiment analysis. *Future Generation Computer Systems*, 113, 58-69.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualised 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)*, pages 2227–2237.

Rosenthal S, Nakov P, Kiritchenko S, Mohammad SM, Ritter A, Stoyanov V. SemEval-2015 task 10: Sentiment Analysis in Twitter. In: Proc. 9th Intl. Workshop on Semantic Evaluation (SemEval); 2015. p. 451–463.

Saif H, Fernández M, He Y, Alani H. Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold. In: *1st Intl. Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM)*; 2013.

Anders Søgaard, Ivan Vulić, Sebastian Ruder, and Manaal' Faruqui. 2019. *Cross-Lingual Word Embeddings*. Morgan & Claypool Publishers.

Matej Ulčar, and Marko Robnik-Šikonja. 2020. FinEst BERT and CroSloEngual BERT. In *International Conference on Text, Speech, and Dialogue, TSD2020*, pp. 104-111.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luoto-lahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *arXiv preprint* arXiv:1912.07076.

Wehrmann, J., Becker, W., Cagnini, H. E., & Barros, R. C. (2017). A character-based convolutional neural network for language-agnostic Twitter sentiment analysis. In 2017 *International Joint Conference on Neural Networks (IJCNN)*, pp. 2384-2391.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2019. Large batch optimisation for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations, ICLR 2019*.

Ranasinghe, T., & Zampieri, M. (2020). Multilingual Offensive Language Identification with Cross-lingual Embeddings. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 5838-5844).

Appendix C: MICE: Mining Idioms with Contextual Embeddings

MICE: Mining Idioms with Contextual Embeddings

Tadej Škvorc

University of Ljubljana, Faculty of Computer and Information Science, 1000 Ljubljana, Slovenia

Jožef Stefan Institute, Jamova Cesta 39, 1000 Ljubljana, Slovenia

Polona Gantar

University of Ljubljana, Faculty of Arts, 1000 Ljubljana, Slovenia

Marko Robnik-Šikonja

University of Ljubljana, Faculty of Computer and Information Science, 1000 Ljubljana, Slovenia

Abstract

Idiomatic expressions can be problematic for natural language processing applications as their meaning cannot be inferred from their constituting words. A lack of successful methodological approaches and sufficiently large datasets prevents the development of machine learning approaches for detecting idioms, especially for expressions that do not occur in the training set. We present an approach, called MICE, that uses contextual embeddings for that purpose. We present a new dataset of multi-word expressions with literal and idiomatic meanings and use it to train a classifier based on two state-of-the-art contextual word embeddings: ELMo and BERT. We show that deep neural networks using both embeddings perform much better than existing approaches, and are capable of detecting idiomatic word use, even for expressions that were not present in the training set. We demonstrate cross-lingual transfer of developed models and analyze the size of the required dataset.

Keywords: Machine learning, Natural language processing, Idiomatic

Email addresses: tadej.skvorc@fri.uni-lj.si (Tadej Škvorc), apolonija.gantar@guest.arnes.si (Polona Gantar), marko.robnik@fri.uni-lj.si (Marko Robnik-Šikonja)

Preprint submitted to Knowledge-Based Systems *November 27, 2020*

expressions, Word embeddings, Contextual embeddings, Cross-lingual transfer

1. Introduction

Idiomatic expressions (IEs), also called idioms, are composed from a group of words whose meaning is established by convention and cannot be deduced from individual words composing the expression (e.g., it's a piece of cake). In this work we are interested in detection and identification of IEs.

Due to the lack of satisfactory tools, linguists often create lexicons of idioms manually or by using tools that take into account only co-occurrence features, since these are easier to implement and are relatively language independent. This type of workflow introduces several problems. First, manually created large lexicons of idioms are scarce because of the time-consuming human labor that is required, particularly for less-resourced languages. Second, frequency lists of idioms that were created without robust, generalized identification tools are unreliable – mostly due to their discontinuity and syntactic variability. Finally, discovery or detection of new IEs is often based on personal knowledge of linguists or frequent collocations. This may completely omit many idioms.

IEs such as "break the ice" and "under the weather" commonly occur in texts. They can be hard to understand for computer models as their meaning differs from the meaning of individual words. To address this, several automatic machine learning based approaches for detection of idiomatic language emerged. However, current approaches suffer from a number of issues and limitations related to methodological shortcomings as well as a lack of datasets. The first issue that affects current approaches is the lack of large datasets with annotated IEs. Because of a large number of different IEs, a dataset that would contain sufficient number of examples for every IE needed to train a classification model currently does not exist. Additionally, most existing datasets only address English, which makes developing approaches for other languages difficult. Existing works use small datasets, such as the data from SemEval 2013, task 5B [1], PARSEME Shared Task on Automatic Verbal Multi-Word Expression (MWE) Identification

[2], or the VNC tokens dataset [3]. These datasets only cover a limited number of IEs and contain at most a few annotated sentences for each expression, which makes it hard to train successful machine-learning models for IE recognition.

Deep neural networks are currently the most successful machine learning approach for textual data, surpassing all other approaches in practically all language processing and understanding tasks [4, 5, 6, 7, 8]. As input, neural networks require numerical data, and texts are transformed into numeric vectors via a process called text embedding. The process has to ensure that relations between words are reflected in distances and directions in a numeric space of typically several hundred dimensions. The embedding vectors are obtained from specialized learning tasks, based on neural networks, e.g., word2vec [9], GloVe [10], or fastText [11]. For training, the embedding algorithms use large monolingual text corpora and design a learning task that tries to predict a context of a given word. The problem of the first generation of neural embeddings, such as word2vec, is their failure to express polysemous words. During training of the 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 about their contexts in proportion to their frequency in the training corpus. This causes the final vector to be placed somewhere in the weighted middle of all word's meanings. Consequently, rare meanings of words (which mostly include their idioms) are poorly expressed with these embeddings 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].

The idea of contextual embeddings is to generate a different vector for each context a word appears in and the context is typically defined sentence-wise. 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 use currently

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

the most successful approaches to contextual word embeddings, ELMo [7] and BERT [8]. We examine whether contextual word embeddings can be used as a solution to the idiom identification problem. Past work shows that contextual word embeddings are capable of detecting different meanings of polysemous words and can improve the performance on a variety of NLP tasks [8]. However, to the best of our knowledge, current approaches have not used contextual word embeddings for differentiating between idiomatic and literal language use. In the proposed approach, called MICE (Mining Idioms with Contextual Embeddings), we use ELMo and BERT embeddings as an input to a neural network and show that using them as the first layer of neural networks improves results compared to existing approaches. We evaluate our approach on a new dataset of Slovene IEs, as well as on the existing dataset from the PARSEME Shared Task on Automatic Verbal MWE Identification. We analyze different properties of the proposed models, such as the amount of labelled data required to get useful results, different variants of BERT models, and cross-lingual transfer of trained models.

We show that contextual embeddings contain a large amount of lexical and semantic information that can be used to detect IEs. Our MICE approach outperforms existing approaches that do not use pre-trained contextual word embeddings in detection of IE present in the training data, as well as identification of IE missing in the training set. The later is a major problem of existing approaches. Finally, we show that multilingual contextual word embeddings are capable of detecting IEs in multiple languages even when trained on a monolingual dataset.

The reminder of the paper is structured as follows. In Section 2, we describe past research on automatic IE detection. We present our MICE methodology in Section 3. Section 4 describes the datasets used for the evaluation of our approach, which we describe in Section 5. Section 6 concludes the paper.

2. Related Work

There currently exists a variety of approaches for detecting IEs in a text, broadly divided into supervised and unsupervised methods. In supervised approaches, the problem is frequently presented as a binary classification problem where a separate classifier is trained for each idiom [12]. The disadvantage of this approach is that it scales poorly to a large number of idioms as it requires a separate training set for each idiom.

In recent years, several neural network approaches have been proposed. MUMULS [13] uses a neural network with a bidirectional gated recurrent units (GRUs) [14] in combination with an embedding layer. In addition to idioms, it is capable of detecting different types of verbal multi-word expressions, which were annotated within the PARSEME Shared Task on Automatic Verbal MWE Identification [2]. MUMULS achieved best results on multiple languages, but the authors reported a poor classification accuracy on languages with a low amount of training data and were unable to detect expressions that did not occur in the training set. The 2018 edition of the shared task [15] featured several other systems based on neural networks [16, 17, 18] with similar outcomes to MUMULS, namely good results on several languages but low classification accuracy and F_1 score for languages with small training datasets and no detection of expressions that are not present in the training set. Another approach was presented by Boroş and Burtica [18], who use a bidirectional long short-term memory network (biLSTM) in combination with graph-based decoding. However, despite using neural networks, these approaches do not use pretrained contextual embeddings. Because of this, they cannot make use of un-annotated datasets when training their model and cannot and makes it more difficult for them to make full use of contextual information in text.

The second broad group of methods for detecting idiomatic word use are unsupervised approaches. Sporleder and Li [19] use lexical cohesion to detect IEs without the need for a labeled dataset or language resources such as dictionaries or lexicons. Liu and Hwa [20] compare the context of a word's occurrence to a

predefined "literal usage representation" (i.e. a collection of words that often appear near literal uses of the word) to obtain a heuristic measure indicating whether a word was used literally or idiomatically. The obtained scores are passed to a probabilistic latent variable model, which predicts the usage of each word. They report average F_1 scores between 0.72 to 0.75 on the SemEval 2013 Task 5B [1] and VNC tokens datasets [3]. This is lower than the results obtained by our model on a comparable task.

A potential problem with current approaches is a lack of large annotated datasets that could be used to train classification models. Liu and Hwa [12] use the data from SemEval 2013 Task 5B [1], which only contains 10 different idioms with 2371 examples. Boroş and Burtica [18] and Klyueva et al. [13] trained their models on the PARSEME Shared Task on Automatic Verbal MWE Identification [2], which only contains a small number of idioms across 20 languages. Larger datasets exist, such as the VNC tokens dataset [3], which contains 2,984 instances of 53 different expressions, and the dataset presented by Fadaee et al. [21], which contains 6,846 sentences with 235 different IEs in English and German. In our work, we use a larger dataset with 29,400 sentences and 75 different IEs.

Existing classification approaches require a list of idiomatic phrases with accompanying datasets on which a classifier is trained. Current approaches pay little attention to detecting idioms that do not appear in the training set, which is a much harder problem. However, due to a large number of idiomatic phrases, such a use is more reflective of real-world problems. Even the unsupervised approach presented by Liu and Hwa [20] first manually constructs literal usage representations for each idiomatic phrase and is therefore not suitable for detecting non-listed IEs. We use contextual embeddings, which can capture semantic information without requiring labelled data for training. This allows then to detect idiomatic phrases even if they do not appear in a pre-defined list.

3. Detecting IEs with Contextual Word Embeddings

We first describe two state-of-the-art deep neural network approaches to contextual embeddings, ELMo [7] and BERT [8], followed by the proposed MICE neural network architectures for identification of IEs.

3.1. ELMo contextual embeddings

ELMo (Embeddings from Language Models) [7] is a large pretrained neural language model, producing contextual embeddings and state-of-the-art results in many text processing tasks. The ELMo architecture consists of three layers of neurons. The output of neurons after each layer gives one set of embeddings, altogether three sets. The first layer is the convolutional (CNN) layer operating on the character-level input. This layer is followed by two biLSTM layers, that consist of two concatenated LSTM layers. The first, left-to-right LSTM layer is trained to predict the following word based on the given past words, where each word is represented by the embeddings from the CNN layer. The second, right-to-left LSTM predicts the preceding word based on the given following words. Although ELMo is trained on character-level input and is able to handle out-of-vocabulary words, a vocabulary file containing the most common tokens is used for efficiency during training and embedding generation.

In NLP tasks, usually a weighted average of the three embeddings is used. The weights for merging the representation of layers are learned during the training of the model for a specific task. Optionally, the entire ELMo model can be fine-tuned for the specific task.

In our work, we use the ELMo model that was pre-trained on a large amount of Slovene text [22]. We take an average of the three ELMo embedding layers as the input to our prediction models. These embeddings are not fine-tuned to the specific task of idiom detection, as we wanted to evaluate how well the embeddings capture the relevant contextual information without task-specific fine-tuning. As results show, even without fine-tuning, the contextual embeddings improve performance compared to similar approaches that do not use contextual

word-embeddings. Fine-tuning of embeddings layer of neural networks is left for further work.

3.2. BERT contextual model

BERT (Bidirectional Encoder Representations from Transformers) [8] generalises the idea of language models to masked language models—inspired by Cloze (i.e. gap filling) tests—which test the understanding of a text by removing a certain portion of words that the participant is asked to fill in. The masked language model randomly masks some of the tokens from the input, and the task of the language model is to predict the missing token based on its neighbourhood. BERT uses transformer architecture of neural networks [23], which uses both left and right context in predicting the masked word 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 result of pre-trained tokenization is that some common words are kept as single tokens, while 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, called mBERT is trained on 104 languages simultaneously. BERT has shown excellent performance on 11 NLP tasks: 8 from GLUE language understanding benchmark [24], question answering, named entity recognition, and common-sense inference.

Rather than training an individual classifier for every classification task from scratch, which would be resource and time expensive, the pre-trained BERT language model is usually used and fine-tuned on a specific task. This approach is common in modern NLP, because large pretrained language models extract highly-relevant textual features without task specific development and training. Frequently, this approach also requires less task-specific data. During pre-training, BERT model learns relations between sentences (entailment) and between tokens within a sentence. This knowledge is used during training on a specific down-stream task [8]. The use of BERT for a token classification task requires adding connections between its last hidden layer and new neurons

corresponding to the number of classes in the intended task. To classify a sequence, we use a special [CLS] token that represents the final hidden state of the input sequence (i.e the sentence). The predicted class label of the [CLS] token corresponds to the class label of the entire sequence. 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.

In our use of BERT models, we did not fine-tune the embedding weights but left them as they were after the original pretraining. This simplification significantly reduces the computational load but leads to potential loss of accuracy. This is a possible improvement to be tested in future work, as fine-tuning the embeddings would likely improve the results.

3.3. The proposed MICE architecture

Our approach is based on contextual word embeddings, which were designed to deal with the fact that a word can have multiple meanings. Instead of assigning the same vector to every occurrence of a word, contextual embeddings assign a different vector to each word occurrence based on its context. As the contexts of words' literal use and idiomatic occurrences of the same word are likely to differ, these embeddings shall be well-suited for detecting IEs. We used two state-of-the-art embedding approaches: ELMo [7] and BERT [8]. For ELMo, we used the pretrained Slovene model described by [22]. The model was trained on the Gigafida corpus [25] of Slovene texts. For BERT embeddings, we use two different models:

1. The multilingual mBERT model presented by Devlin et al. [8], which was trained on Wikipedia text from 104 languages, including Slovene.
2. The trilingual CroSloEngual BERT presented by Ulčar and Robnik-Šikonja [26], which was trained on English, Slovene, and Croatian using Wikipedia for English text, the Gigafida corpus for Slovene text, and a combination of hrWaC [27], articles from the Styria media group, and Riznica corpora [28] for Croatian text. This BERT is better suited for classification tasks

in Slovene and Croatian as mBERT as its training incorporated larger amounts of training data and a larger vocabulary for each of the involved languages. The authors also report improved cross-lingual transfer of trained models between the three languages.

We use the embeddings (ELMo or BERT) as the first layer of a neural network. This layer is followed by a bidirectional gated recurrent unit (GRU) with 100 cells. GRUs are similar to standard recurrent units but use an additional update and reset gate to help deal with the vanishing gradient problem. The update gate is defined as

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1} + b_z), \tag{1}$$

where $W^(z)$ and $U^(z)$ are trainable weights, x_t is the input vector and b_z is the trainable bias. h_{t-1} represents the memory of past inputs computed by the network. The reset gate uses the same equation, with different weights and biases:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1} + b_r). \tag{2}$$

For each input, the GRU computes the output as:

$$h_t = z_t \odot h_{t-1} + (1 - z) \odot \tanh(W^{(h)}x_t + U^{(h)}(r_t \odot h_{t-1}) + b_h), \tag{3}$$

where $\odot$ is the Hadamard product, and $W^{(h)}$, $U^{(h)}$, and b_h are trainable weights and biases.

For both embeddings used, ELMo and BERT, we follow the GRU layer with a softmax layer for obtaining the final predictions. A dropout of 50% is applied at the softmax layer. This approach follows the work on MWE detection presented by Klyueva et al. [13] but with the difference that we use contextual embeddings. We deliberately use a simple network architecture to show that the embeddings, by themselves, capture enough semantic information to properly recognize IEs.

We use the architecture on two types of classification tasks: a token-level classification, where we predict whether an individual token has an idiomatic or literal meaning, and a sentence-level classification, where the network makes a single prediction for the entire sentence, predicting whether the sentence contains

an expression with an idiomatic meaning. The details of the tasks are presented in Section 5.

We fine-tuned the hyperparameters using a development set consisting of 7% of sentences randomly selected from our dataset, as described in Section 4.1. We trained the network for 10 epochs using RMSProp as the optimizer with the learning rate of 0.001, $\rho = 0.9$, and $\epsilon = 10^{-7}$. We used the binary cross-entropy as the loss function.

4. Datasets

Our approach supports two types of tasks, monolingual and multilingual. The monolingual approach requires a reasonably large dataset with a sufficient number of idioms. We analyze the required size of a dataset in terms of different idioms and examples of their usage in both monolingual and multilingual settings in Section 5.5. The multilingual approach exploits the existing monolingual dataset to transfer the trained model to languages with fewer resources, i.e. with non-existent or smaller datasets.

In Section 4.1 we describe our monolingual Slovene dataset. In Section 4.2 we describe the well-known PARSEME datasets [2] for detection of multi-word expressions in many languages, which also include idioms.

4.1. Monolingual dataset

We evaluate our approach on a new dataset of Slovene IEs, called SloIE, which we make publicly available for further research[2]. The dataset consists of 29,400 sentences extracted from the Gigafida corpus [25] and contains 75 different IEs. The 75 IEs were selected from the Slovene Lexical Database [29] and had to meet the condition that they appear in corpus sentences in both idiomatic and literal senses, such as, e.g., break the ice, step on someone's toes. Manual selection of idiomatic examples showed that most IEs in the Slovene Lexical Database (2,041 in total) appear more frequently or even exclusively

[2]http://hdl.handle.net/11356/1335

in their idiomatic meaning, either because literal use is not possible (e.g., get under someone's skin), or it's very rare, although possible in terms of syntax and semantics (e.g. to do something behind someone's back). Although this finding is interesting from a (socio)linguistic point of view, in designing the dataset for our purposes, we assumed that the literal and idiomatic interpretation of an expression can be disambiguated by its context.

Two annotators, students of linguistics, marked the complete set of 29,400 sentences. They had four possible choices: YES (the expression in a particular sentence is used in the idiomatic sense, NO (the expression is used in the literal sense), DON'T KNOW (not sure whether the expression is used in a literal or idiomatic sense) and VAGUE, (literal or idiomatic use cannot be inferred from the sentence). Student annotators were previously briefed with short instructions and provided with a sample of good examples. For the training of classification models, we selected only sentences where both annotators agreed on the annotation. The inter-annotator agreement across the entire dataset was 0.952.

Due to the nature of IEs, our dataset is imbalanced. A few expressions occur proportionally in both literal and idiomatic use, while most expressions occurring predominately idiomatically. The dataset contains fewer than 100 occurrences for most expressions. Table 1 shows an overview of the data present in our dataset. The distribution of literal and idiomatic uses of each expression is shown in Figure 1.

SloIE is much larger than other existing datasets in terms of number of sentences, e.g., VNC tokens contains 2984 instances of 53 IEs. Such a dataset would require significant effort to create for other languages. For that reason, we analyze the size and distribution required for successful IE detection models in Section 5.5.

4.2. PARSEME datasets

The dataset for the Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions (MWEs) consists of 280,838 anno-

Table 1: An overview of the data present in the SloIE dataset.

Sentences	29,400
Tokens	695,636
Idiomatic sentences	24,349
Literal sentences	5,051
Idiomatic tokens	67,088
Literal tokens	626,707
Different IEs	75

tated sentences split across 20 languages. The corpus contains annotations for various types of verbal MWEs, such as verb-particle constructions, inherently reflexive verbs, and verbal idioms. As our work focuses on detecting IEs, we only predict tags of verbal idioms. A summary of the number of sentences for each language used in our work is presented in Table 2. We do not use the Arabic dataset as it was not made available under an open licence.

IEs in the PARSEME datasets only occur in a small number of sentences. Additionally, most IEs occur only once in the corpus, which makes training a classifier difficult. For that reason, we used the PARSEME dataset to evaluate our cross-lingual model. The model used the pretrained mBERT embeddings from [8], was further trained on our Slovene SloIE dataset, and tested on each of the PARSEME datasets in different languages. The details are reported in Section 5.4.

5. Evaluation

We evaluate our MICE approach in five different settings, explained below. We present the results of these evaluation scenarios in the subsections.

1. *Classification of IEs that were present in the training set.* In Section 5.1 we evaluate whether MICE is capable of detecting IEs that were present in the training set. We split this task into two sub-tasks: i) sentence-level

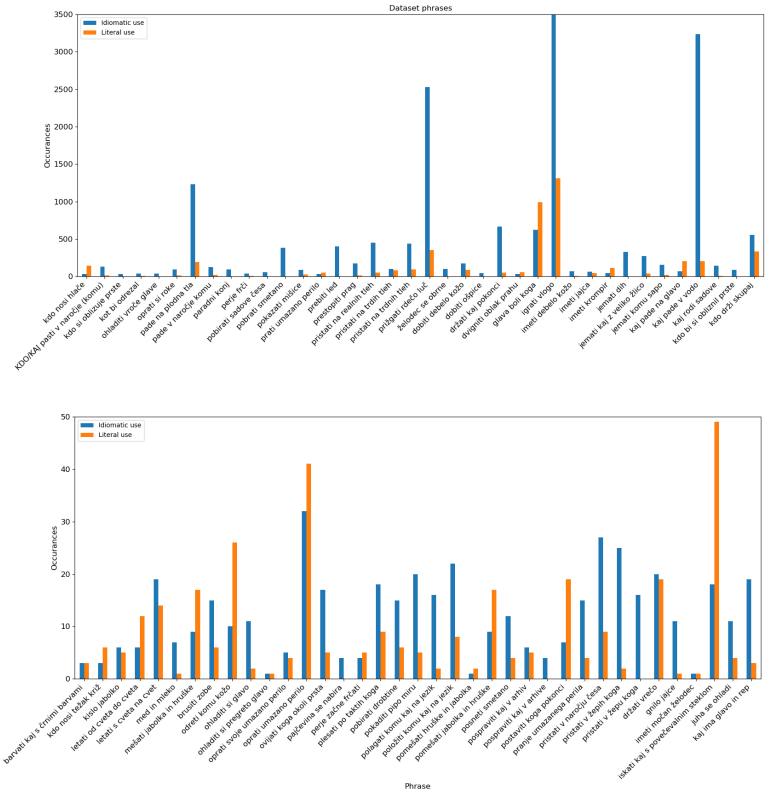


Figure 1: The number of literal and idiomatic uses for IEs present in the SloIE dataset. The top figure shows IEs that occur more than 35 times with an idiomatic meaning. The bottom figure shows IEs that occur less than 35 times with an idiomatic meaning.

classification, where the network makes a single prediction for the entire sentence, predicting whether that sentence contains an expression with an idiomatic meaning and ii) token-level classification, where we predict whether each token has a literal or idiomatic meaning. The sentence-level classification task is easier, but the token-level task can be more useful, as it can be used to detect which tokens have the idiomatic meaning.

2. *Classification of IEs that were not present in the training set.* Due to a large number of idioms, it is difficult and expensive to annotate a dataset

Table 2: An overview of the data present in the PARSEME datasets. Of the 20 languages in the PARSEME corpus, we use 18. We omit Arabic because it is not available as an open language and Farsi, which does not contain IEs. On average, each language contains 586 IEs.

Language	Sentences	Tokens	IEs
BG	6,913	157,647	417
DE	6,261	120,840	1,005
EL	5,244	142,322	515
EN	7,436	124,203	59
ES	2,502	102,090	196
FA	2,736	46,530	0
FR	17,880	450,221	1,786
HE	4,673	99,790	86
HU	3,569	87,777	92
HR	3003	69915	131
IT	15,728	387,325	913
LT	12,153	209,636	229
MT	5,965	141,096	261
PL	11,578	191,239	317
PT	19,640	359,345	820
RO	45,469	778,674	524
SL	8,881	183,285	283
SV	200	3,376	9
TR	16,715	334,880	2,911
Total	19,6546	3,990,191	10,554

that would cover every idiom. Because of this, it would be desirable that the prediction model is capable of detecting expressions that are not present in the training set. We test this setting in Section 5.2. As with the first task, we use sentence-level and token-level classification. This task is more difficult than detection of IEs present in the dataset, and can only be

solved successfully if the contextual word embeddings contain information about idiomatic word use (e.g., as directions in the vector space).

3. *Difference in detection of individual IEs.* It is possible that success in detection of different IEs differs significantly, where some IEs are easy and other much more difficult to detect. In Section 5.3 we evaluate how well our model detects each IE in our dataset and present the differences.
4. *Cross-lingual transfer on the PARSEME dataset.* In Section 5.4 we evaluate whether our approach can be used to detect expressions in different languages when trained with multilingual word embedding models. For testing this hypothesis, we use 18 languages from the multilingual PARSEME dataset containing.
5. *Required size of a dataset.* Our dataset is significantly larger than other datasets used for automatic idiom detection, e.g., PARSEME (for a single language). In Section 5.5 we conduct a series of experiments that provide an information how large dataset (in terms of number of IE and number of examples per IE) is actually needed for successful detection of idioms. This information may be valuable for other languages where similar detection tools will be built.

We compare the proposed MICE approach to different existing approaches. As a baseline, we use the SVM classifier with the tf-idf weighted vector of a sentence as the input. We compare our approach to MUMULS [13], which uses a similar neural network architecture to our approach but does not use pretrained contextual word embeddings. Unlike our approach, MUMULS uses part-of-speech tags and word lemmas as additional inputs.

For all tests, we report the classification accuracy (CA) and F_1 score. As many of the tasks are highly imbalanced, CA is not a good measure and we mostly use the obtained F_1 scores in interpretations of results.

5.1. IEs from the training set

For the first experiment, detection of IEs present in the training set, we randomly split the SloIE dataset into training, testing, and development sets

with the ratio of 63:30:7 (18,522, 8,820, and 2,058 sentences). The network was trained for 10 epochs using RMSProp as the optimizer with a learning rate of 0.001, $\rho = 0.9$, and $\epsilon = 10^{-7}$. Binary cross-entropy was used as the loss function. We report two sets of results: recognition of individual tokens in a sentence as idiomatic or non-idiomatic (i.e. token-level classification), and detection of the whole sentence as either containing or not containing idioms (i.e. sentence-level classification).

The results for token-level classification are presented in Table 3. To provide a sensible context for token-based classification, the input of the SVM classifier consists of the target token and three words before and three words after the target word. The SVM classifier obtains better F_1 score than MUMULS but lower score compared to MICE variants. The dataset is highly imbalanced, with 96,7% of all tokens being non-idiomatic. Lacking discriminating information, MUMULS predicts almost every token as non-idiomatic, which results in high classification accuracy but a very low F_1 score. Due to the imbalanced nature of the dataset, the F_1 score is more reflective of relevant real-world performance, and here the MICE variants are in the class of their own.

Table 3: Comparison of results when classifying tokens with the same IEs present in the training and testing set. Each token was classified as either belonging to IE with the literal meaning, belonging to IE with the idiomatic meaning, or not belonging to IE.

Method	CA	F_1
Default classifier	0.903	0.176
SVM baseline	0.8756	0.3962
MUMULS	**0.975**	0.0659
MICE with Slovene ELMo	0.889	**0.9219**
MICE with mBERT	0.814	0.4556
MICE with CroSloEngual BERT	0.972	0.837

Of the three MICE approaches, the one with the Slovene ELMo model obtains the highest F_1 score. The MICE variants with BERT embeddings obtain

lower classification accuracies and F_1 scores. This is likely due to different tokenization approaches used by the embeddings. We used ELMo embeddings by first performing word-level tokenization while BERT splits words into sub-word units. Token-level classification with BERT must classify sub-word units instead of classifying entire words, as is the case with ELMo. Additionally, our ELMo embeddings were pretrained on a large amount of only Slovene texts, while the mBERT model was trained on 104 different languages. Only a small amount of Slovene texts was included in its training and it has a small proportion of Slovene words in the vocabulary. The CroSloEngual embeddings were trained on a larger amount of Slovene text and therefore achieve better results.

In evaluation on the sentence-level, instead of classifying each token, we classified each sentence whether it contains a IE or not. This lowers the importance of different tokenization strategies between ELMo and BERT. However, sentence-level evaluation does not show whether the approaches are capable of detecting specific words in a sentence as idioms. The results of this evaluation are presented in Table 4.

Table 4: Comparison of results when classifying sentences from the SloIE dataset and the same IEs are present in the training and testing sets. Each sentence was classified as either containing an expression with the literal meaning or containing an expression with the idiomatic meaning.

Method	CA	F_1
Default classifier	0.828	0.906
SVM baseline	0.900	0.942
MUMULS	0.915	0.948
MICE with Slovene ELMo	**0.951**	**0.980**
MICE with mBERT	0.897	0.908
MICE with CroSloEngual BERT	0.921	0.954

The sentence-level classification task is less difficult, which leads to an improved performance for all models. The SVM baseline outperforms the mBERT model. MUMULS also achieves better results, outperforming the SVM baseline

and the mBERT approach. MICE with CroSloEngual BERT is closer to ELMo in this task, though the latter still achieves the best scores. MICE with mBERT likely achieves lower scores because this model was not pretrained on a large enough amount of Slovene text.

5.2. IEs outside the training set

In the previous experiment with the same IEs present in both the training and testing set, we were able to obtain good results (especially with our contextual embeddings approach). However, many languages lack large annotated datasets and even when they do exist, they are unlikely to contain every possible IE found in that language. Because of this, evaluations containing IEs in both sets over-estimates the practical importance of tested methods.

To address this, we tested how well the approaches based on contextual word embeddings generalize to IEs outside the training set. For this experiment, we split our dataset into a training and testing set so that IEs from the testing set do not appear in the training set. Apart from this change, everything else remained the same as in section 5.1 above.

Since IEs in the test set are not present in the training set, the classification models cannot learn how to detect them based on word-data alone. We hypothesize that their detection is possible based on contexts in which they appear. As the meaning of an IE is different from the literal meaning of its constituting words, it should appear in a different context. Neural networks with contextual word embeddings could detect such occurrences. Indeed, our results for token- and sentence-level IE detection, presented in Tables 5 and 6, show that approaches that do not use contextual word embeddings fail to successfully detect IEs that did not occur in the training set, while MICE approaches using contextual embeddings extract useful information.

For token level results, shown in Table 5, due to the imbalanced class distribution, all approaches lag behind the default classifier concerning CA. For both the SVM baseline and MUMULS this is the case also in terms of F_1 score. The MICE approach with ELMo and mBERT models manages to correctly

classify a number of IEs, though the results are worse than in the scenario, where the same IEs are present in both the training and testing set. MICE with ELMO embeddings is again the best method, while CroSloEngual embeddings are surprisingly unsuccessful.

Table 5: Comparison of results when classifying tokens and test set IEs are not present in the training set.

Method	CA	F_1 score
Default classifier	**0.903**	0.176
SVM baseline	0.870	0.029
MUMULS	0.873	0.000
MICE with Slovene ELMo	0.803	**0.866**
MICE with mBERT	0.733	0.803
MICE with CroSloEngual BERT	0.759	0.176

Sentence-level results in Table 6 show improved scores of all models. The SVM baseline and MUMULS still lag behind the default classifier concerning both CA and F_1 score. MICE approaches are better, with Slovene ELMo variant achieving the best scores.

Table 6: Comparison of results when classifying sentences and the test set IEs are not present in the training set.

Method	CA	F_1 score
Default classifier	0.828	0.906
SVM baseline	0.783	0.689
MUMULS	0.520	0.672
MICE with Slovene ELMo	**0.842**	**0.907**
MICE with mBERT	0.836	0.904
MICE with CroSloEngual BERT	0.771	0.837

5.3. Evaluation of individual IEs

In addition to cumulative results of the entire test set, we are also interested in individual differences between IEs, as it is possible that some IEs are easy and others are hard to detect. As the meanings of IEs can vary from being similar or very different to the literal meanings of their words, we assume that the ability of models based on contextual word embeddings could vary significantly. For this task, we train the detection models on all other IEs (74 of them) and test them on the left-out IE. In this way, we obtain a separate detection model for each IE, trained on every sentence that did not contain that IE, and evaluate it on the sentences containing that IE (similar out-of-test-set sentence-level scenario as in Section 5.2). For this evaluation, we used the MICE Slovene ELMo model described in Section 5.2, as it outperformed all other models in previous tests.

Figure 2 shows the distribution of F_1 scores across all the IEs in our SloIE dataset. The distribution shows that for the majority of IEs, MICE models achieve high F_1 scores above 0.8, while there are a few IEs with low recognition rate with $F_1 < 0.6$. In Table 7 we elaborate on these results and show the five best and worst recognizable IEs. At the moment, we do not have an interpretation why certain IEs are more or less difficult to detect, and leave this question for further work.

5.4. Cross-lingual evaluation of IEs

The results above show encouraging results for IE detection in a language with sufficiently large datasets. As recent research on cross-lingual embeddings shows that reasonably good transfer of trained models can be obtained for many tasks [30, 31, 32, 33], we attempt such a transfer of our models. We use the dataset from the PARSEME shared task on automatic identification of verbal multiword expressions described in Section 4.2. We evaluated two contextual embeddings discussed in the previous sections: the Slovene ELMo embeddings and the multilingual BERT embeddings. We evaluated the cross-lingual MICE approach in the following manner:

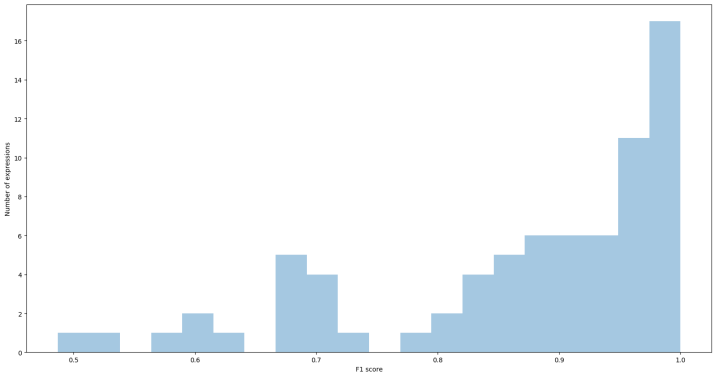


Figure 2: The distribution of F_1 scores per IEs in sentence-level task on the out-of-test-set task using MICE with Slovene ELMo embeddings.

- We evaluated MICE with Slovene ELMo embeddings on Slavic languages similar to Slovene, with datasets present in the PARSEME collection, i.e., Slovene, Croatian, and Polish. As the Slovene ELMo embeddings are not multilingual, they are unlikely to generalize to other languages. In future work, we plan to use these embeddings for prediction in other languages by using cross-lingual mappings (e. g., [34]).

- We evaluated MICE with mBERT embeddings on all languages from the PARSEME collection. The mBERT model was trained on 104 languages, including every language present in the PARSEME dataset.

For both test-cases, we constructed balanced datasets which consist of every sentence with IEs from the PARSEME dataset in a given language, and an equal number of sentences without IEs, chosen at random from the same dataset. We performed the evaluation on the sentence-level classification task.

For the Slavic languages test, we trained the prediction model on the whole SloIE dataset, presented in Section 4.1. We did not train the model on any multilingual data to see whether the contextual embeddings alone are enough to generalize to other languages, at least to similar ones such as Croatian. For all

Table 7: Examples of the easiest and most difficult IEs for MICE with Slovene ELMo embeddings.

IE	F_1 score	Number of detected IEs
Pospraviti v arhive	1.0	4
Kislo jabolko	1.0	9
Pomešati jabolka in hruške	1.0	33
Pristati v žepih nekoga	1.0	28
Perje začne frčati	1.0	19
Pospraviti kaj v arhiv	0.600	12
Imeti krompir	0.597	162
Gnilo jajce	0.571	11
Kdo nosi hlače	0.525	218
Želodec se obrne	0.487	10

PARSEME languages using MICE with mBERT, we split each dataset into the training, testing and validation sets using a 60:30:10 ratio, trained the model for each language on the training set and evaluated it on the testing set. For Slovene, Croatian, and Polish we additionally trained MICE mBERT models on the SloIE dataset, as the similarity of those languages means that additional data in the Slovene language could be beneficial. The results are presented in Table 8.

The results of the monolingual evaluation presented in Section 5.2 are also confirmed on the Slovene PARSEME dataset, as MICE with Slovene ELMO model is capable of detecting idioms in that dataset. The same model generalizes very well to the PARSEME Croatian dataset, likely due to its similarity to Slovene. The generalization to Polish, which is more distant Slavic language, is not successful. MICE models with mBERT also generalize well for a few languages. They obtain good results on Slovene and Croatian, likely due to the large amount of training data in the SloIE corpus, which also generalizes to Croatian idioms. The MICE mBERT models outperform default classifiers in

Table 8: Results of the multilingual evaluation. The MICE models with Slovene ELMo embeddings were evaluated on Slavic languages similar to Slovene, while the variants with mBERT were tested for all languages in PARSEME dataset which contain IEs. We report F_1 scores and include default classifiers as a reference.

Language	Slovene ELMo	mBERT	Default F_1
Slovene	0.8163	0.8359	0.667
Croatian	0.9191	0.8970	0.667
Polish	0.2863	0.6987	0.667
English	-	0.650	0.667
French	-	0.814	0.667
German	-	0.622	0.667
Turkish	-	0.682	0.667
Romanian	-	0.625	0.667
Lithuanian	-	0.689	0.667
Italian	-	0.683	0.667
Hungarian	-	0.555	0.667
Hindi	-	0.562	0.667
Hebrew	-	0.693	0.667
Farsi	-	-	-
Basque	-	0.692	0.667
Spanish	-	0.340	0.667
Greek	-	0.484	0.667
Bulgarian	-	0.601	0.667

French, Turkish, Lithuanian, Italian, Hebrew, and Basque, despite small amounts of training data, low numbers of IEs in training sets, most IEs only appearing once, and IEs in the testing set not appearing in the training set. They perform less well on other languages but are still capable of detecting some IEs.

MUMULS and the SVM baseline were both unable to detect IEs in other languages, obtaining the F_1 score of 0 in all cases.

5.5. Effect of the dataset size

Most languages currently do not have IE datasets, and it might be helpful to provide an information on how large datasets are required. In this section, we analyze the size of dataset needed to obtain acceptable performance in Slovene language and expect that findings will generalize to other languages. Further, as our SloIE dataset is larger than existing IE datasets, our results are not directly comparable to existing research, which was evaluated on smaller datasets. Our evaluation will shed light on this question as well.

We approach the analysis by running a number of tests on subsets of SloIE dataset. We randomly selected subsets of different sizes (100 %, 80%, 60%, 40%, 20%, and 10% percent) and re-ran the evaluations, repeating tests with IEs from the training set (Section 5.1). We only tested our best model, MICE, with Slovene ELMo embeddings. We show the results when classifying IEs from the training set in Table 9. The results show that MICE performs well even when using smaller datasets. The F_1 score and CA slowly decrease with lower numbers of training sentences and remain quite high even with smaller training sets. This means that our approach could achieve good real-world performance even with languages that do not have large annotated datasets. When classifying IEs from outside the training set, the results did not significantly change with lower dataset sizes.

Our final evaluation checks whether a balanced dataset improves the result. The SloIE dataset is highly imbalanced (both in the number of examples per IE and in the number of idiomatic and literal use cases of each expression). This might make training neural networks difficult. To determine how much the dataset imbalance effects the results we constructed a smaller, balanced dataset, that contains the same amount of idiomatic/non-idiomatic sentences for each expression. The balanced version of the dataset contains 5481 training sentences and 2349 testing sentences across 75 IEs.

The balanced dataset is much smaller than the original dataset, and possibly reduced performance may be due to a less training data. For a more fair comparison, we also constructed a smaller, imbalanced dataset by taking a

Table 9: The effect of dataset size on classification accuracy (CA) and F_1 score using the sentence-level classification task with IEs that appear in the training set, using MICE with Slovene ELMo embeddings.

Sentences	CA	F_1 score
27698	0.903	0.938
17449	0.906	0.0.942
9771	0.902	0.938
4787	0.870	0.934
2010	0.894	0.934
703	0.874	0.924

random subset of SloIE sentences for each expression equal in size to the balanced dataset. The size and number of sentences for the imbalanced dataset was the same as the balanced version.

We performed sentence-level classification on the two datasets, predicting IEs present in the training set. The results of the classification are shown in Table 10. Results show that training the model on the balanced dataset did not lead to an improved classification accuracy or F_1 score. This indicates that MICE is insensitive to this sort of imbalance and performs well even when trained on imbalanced datasets.

Table 10: The effect of using a balanced dataset on classification accuracy and F_1 score. The evaluation was conducted as a sentence-level classification task with IEs appearing in the training set, and using MICE with Slovene ELMo embeddings.

Dataset	CA	F_1 score	Default CA	Default F_1
Balanced	0.8011	0.766	0.500	0.667
Imbalanced	0.812	0.853	0.625	0.767

6. Conclusion and Future Work

We showed that contextual word embeddings can be used with neural networks to successfully detect IEs in text. When contextual embeddings (ELMo or mBERRT) were used as the first layer of a neural network with the same architecture as the existing MUMULS approach, we were able to obtain much better results. While the existing approaches performed well on sentence-level classification of IEs that were present in the training set, they failed on token-level tasks and when detecting new IEs, not present in the training set. We showed that using fine-tuned contextual word embeddings allows the network to perform better on token-level classification and to successfully generalize to IEs that were not present in the training set. This opens an opportunity for successful treatment of IEs in many downstream applications. We published our code and models under the CC licence[3].

We evaluated our MICE approach on SloIE dataset, a new, large dataset of Slovene idioms, as well as on the existing multilingual PARSEME datasets. SloIE dataset, which we made publicly available[4], is larger than most of existing datasets, and should therefore be useful for further research into automatic idiom detection. Additionally, we evaluated how the size of the dataset affects the results and showed that our approaches perform well even when trained on smaller datasets.

We show that contextual word embeddings are capable of generalizing to other languages. When dealing with similar language pairs (e. g., Slovene-Croatian), both the monolingual ELMo embeddings and the multilingual BERT embeddings were capable of detecting idioms in Croatian text when trained only on Slovene. The multilingual BERT model was able to detect idioms even in some more distant languages, though with reduced classification accuracy and F_1 scores.

Our work could be improved and extended in multiple ways. We only used

[3] https://github.com/TadejSkvorc/MICE
[4] http://hdl.handle.net/11356/1335

embeddings that were pretrained on general text and were not fine-tuned for the specific task of detecting idiomatic language. Several authors have shown [35, 8] that specializing embeddings for specific tasks can improve results on a variety of NLP tasks. Several such approaches could be applied to our task and would likely further improve the performance. Additionally, we intentionally used a simple network architecture that could be improved in the future. Finally, to put our models into a practical use, we intend to apply MICE models in the task of IE lexicon construction.

Acknowledgements

The research was supported by the Slovene Research Agency through research core funding no. P6-0411 and P6-0215, as well as project J6-8256 (New grammar of contemporary standard Slovene: sources and methods). This paper is supported by European Union's Horizon 2020 Programme project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media, grant no. 825153). The results of this publication reflects only the authors' view and the Commission is not responsible for any use that may be made of the information it contains.

The SloIE dataset was annotated by student annotators Kaja Žvanut, Tajda Liplin-Šerbetar, Karolina Zgaga and Tjaša Jelovšek. A part of it was also annotated by a non-native speaker Danijela Topić-Vizcaya.

References

[1] I. Korkontzelos, T. Zesch, F. M. Zanzotto, C. Biemann, Semeval-2013 task 5: Evaluating phrasal semantics, in: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), 2013, pp. 39–47.

[2] A. Savary, C. Ramisch, S. Cordeiro, F. Sangati, V. Vincze, B. QasemiZadeh, M. Candito, F. Cap, V. Giouli, I. Stoyanova, A. Doucet, The PARSEME

shared task on automatic identification of verbal multiword expressions, in: Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017), Association for Computational Linguistics, 2017, pp. 31–47.

[3] P. Cook, A. Fazly, S. Stevenson, The VNC-tokens dataset, in: Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008), 2008, pp. 19–22.

[4] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[5] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in neural information processing systems, 2015, pp. 649–657.

[6] Y. Kim, Y. Jernite, D. Sontag, A. M. Rush, Character-aware neural language models., in: AAAI, 2016, pp. 2741–2749.

[7] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of NAACL-HLT, 2018, pp. 2227–2237.

[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), Association for Computational Linguistics, 2019, pp. 4171–4186.

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

[10] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on Empirical methods in natural language processing EMNLP, 2014, pp. 1532–1543.

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

[12] C. Liu, R. Hwa, Representations of context in recognizing the figurative and literal usages of idioms, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 3230–3236.

[13] N. Klyueva, A. Doucet, M. Straka, Neural networks for multi-word expression detection, in: Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017), 2017, pp. 60–65.

[14] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.

[15] C. Ramisch, S. R. Cordeiro, A. Savary, V. Vincze, V. Barbu Mititelu, A. Bhatia, M. Buljan, M. Candito, P. Gantar, V. Giouli, T. Güngör, A. Hawwari, U. Iñurrieta, J. Kovalevskaitė, S. Krek, T. Lichte, C. Liebeskind, J. Monti, C. Parra Escartín, B. QasemiZadeh, R. Ramisch, N. Schneider, I. Stoyanova, A. Vaidya, A. Walsh, Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions, in: Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018), 2018, pp. 222–240.

[16] G. Berk, B. Erden, T. Güngör, Deep-BGT at PARSEME shared task 2018: Bidirectional lstm-crf model for verbal multiword expression identification, in: Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018), 2018, pp. 248–253.

[17] R. Ehren, T. Lichte, Y. Samih, Mumpitz at PARSEME shared task 2018: A bidirectional lstm for the identification of verbal multiword expressions,

in: Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018), 2018, pp. 261–267.

[18] T. Boroş, R. Burtica, GBD-NER at PARSEME shared task 2018: Multiword expression detection using bidirectional long-short-term memory networks and graph-based decoding, in: Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018), 2018, pp. 254–260.

[19] C. Sporleder, L. Li, Unsupervised recognition of literal and non-literal use of idiomatic expressions, in: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), 2009, pp. 754–762.

[20] C. Liu, R. Hwa, Heuristically informed unsupervised idiom usage recognition, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1723–1731.

[21] M. Fadaee, A. Bisazza, C. Monz, Examining the tip of the iceberg: A data set for idiom translation, arXiv preprint arXiv:1802.04681 (2018).

[22] M. Ulčar, M. Robnik-Šikonja, High quality ELMo embeddings for seven less-resourced languages, in: Proceedings of The 12th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 4731–4738.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[24] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, 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, 2018, pp. 353–355.

[25] S. Krek, P. Gantar, Š. A. Holdt, V. Gorjanc, Nadgradnja korpusov Gigafida, Kres, ccGigafida in ccKres, in: Proceedings of Language technologies and digital humanistics, 2016, pp. 200–202.

[26] M. Ulčar, M. Robnik-Šikonja, FinEst BERT and CroSloEngual BERT: less is more in multilingual models., in: Proceedings of Text, Speech, and Dialogue, TSD 2020, 2020. (accepted).

[27] N. Ljubešić, T. Erjavec, hrWaC and slWaC: Compiling web corpora for Croatian and Slovene, in: International Conference on Text, Speech and Dialogue, Springer, 2011, pp. 395–402.

[28] D. Ćavar, D. B. Rončević, Riznica: the Croatian language corpus, Prace filologiczne 63 (2012) 51–65.

[29] P. Gantar, S. Krek, Slovene lexical database, in: Natural language processing, multilinguality: 6th international conference, 2011, pp. 72–80.

[30] S. Ruder, I. Vulić, A. Søgaard, A survey of cross-lingual word embedding models, Journal of Artificial Intelligence Research 65 (2019) 569–631.

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

[32] M. Robnik-Šikonja, K. Reba, I. Mozetič, Cross-lingual transfer of Twitter sentiment models using a common vector space, 2020. URL: `https://arxiv.org/pdf/2005.07456`.

[33] E. Linhares Pontes, J. G. Moreno, A. Doucet, Linking named entities across languages using multilingual word embeddings, in: 20th ACM/IEEE Joint Conference on Digital Libraries, JCDL 2020, 2020.

[34] T. Schuster, O. Ram, R. Barzilay, A. Globerson, Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing, in: Proceedings of the 2019 Conference of the North American Chapter

of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1599–1613. URL: https://www.aclweb.org/anthology/N19-1162. doi:10.18653/v1/N19-1162.

[35] X. L. Li, J. Eisner, Specializing word embeddings (for parsing) by information bottleneck, 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), 2019, pp. 2744–2754.