Character-Level and Syntax-Level Models for Low-Resource and Multilingual Natural Language Processing

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

eingereicht von
Silvia Severini

München, den 25. Oktober 2022

Erstgutachter: Prof. Dr. Hinrich Schütze
Zweitgutachter: Prof. Dr. Danushka Bollegala
Drittgutachter: Dr. Pierre Zweigenbaum

Tag der Einreichung: 25. Oktober 2022
Tag der mündlichen Prüfung: 05. Juli 2023

Eidesstattliche Versicherung
(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5.)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig ohne unerlaubte Beihilfe angefertigt ist.

München, den 25. Oktober 2022.

____
Silvia Severini

Abstract

There are more than 7000 languages in the world, but only a small portion of them benefit from Natural Language Processing resources and models. Although languages generally present different characteristics, “cross-lingual bridges” can be exploited, such as transliteration signals and word alignment links. Such information, together with the availability of multiparallel corpora and the urge to overcome language barriers, motivates us to build models that represent more of the world’s languages.

This thesis investigates cross-lingual links for improving the processing of low-resource languages with language-agnostic models at the character and syntax level. Specifically, we propose to (i) use orthographic similarities and transliteration between Named Entities and rare words in different languages to improve the construction of Bilingual Word Embeddings (BWEs) and named entity resources, and (ii) exploit multiparallel corpora for projecting labels from high- to low-resource languages, thereby gaining access to weakly supervised processing methods for the latter.

In the first publication, we describe our approach for improving the translation of rare words and named entities for the Bilingual Dictionary Induction (BDI) task, using orthography and transliteration information. In our second work, we tackle BDI by enriching BWEs with orthography embeddings and a number of other features, using our classification-based system to overcome script differences among languages. The third publication describes cheap cross-lingual signals that should be considered when building mapping approaches for BWEs since they are simple to extract, effective for bootstrapping the mapping of BWEs, and overcome the failure of unsupervised methods. The fourth paper shows our approach for extracting a named entity resource for 1340 languages, including very low-resource languages from all major areas of linguistic diversity. We exploit parallel corpus statistics and transliteration models and obtain improved performance over prior work. Lastly, the fifth work models annotation projection as a graph-based label propagation problem for the part of speech tagging task. Part of speech models trained on our labeled sets outperform prior work for low-resource languages like Bambara (an African language spoken in Mali), Erzya (a Uralic language spoken in Russia’s Republic of Mordovia), Manx (the Celtic language of the Isle of Man), and Yoruba (a Niger-Congo language spoken in Nigeria and surrounding countries).

Zusammenfassung

Es existieren mehr als 7000 Sprachen auf der Welt, aber nur wenige davon profitieren von Sprachverarbeitungsressourcen und -modellen. Obwohl Sprachen sich in vielerlei Hinsicht unterscheiden, können ßprachübergreifende Brücken"genutzt werden, z. B. Signale aus Transliterationen oder Wort-Alignierungen. Die Verfügbarkeit von solchen Informationsquellen sowie von multiparallelen Korpora und der Wunsch, Sprachbarrieren zu überwinden, motivieren uns, Sprachtechnologie auf viele weitere Sprachen auszuweiten.

In dieser Arbeit untersuchen wir sprachübergreifende Verbindungen, mit denen wir die Verarbeitung von ressourcenarmen Sprachen mit sprachunabhängigen Modellen auf der Buchstaben- und Syntaxebene verbessern können. Konkret schlagen wir vor, (i) orthografische Ähnlichkeiten und Transliteration von Eigennamen (Named Entities; NEs) und seltenen Wörtern in unterschiedlichen Sprachen zu nutzen, um die Erstellung von besseren zweisprachige Worteinbettungen (Bilingual Word Embeddings; BWEs) und Ressourcen für die Named-Entity-Ressourcen zu ermöglichen, und (ii) multiparallele Korpora für die Projektion von Labels von ressourcenreichen zu ressourcenarmen Sprachen zu nutzen, um so Zugang zu schwach überwachten Verarbeitungsmethoden für letztere zu erhalten.

In der ersten Veröffentlichung beschreiben wir unseren Ansatz zur Verbesserung der Übersetzung von seltenen Wörtern und Eigennamen zur automatischen Erstellung zweisprachiger Wörterbücher (Bilingual Dictionary Induction; BDI) unter Verwendung von Orthographie- und Transliterationsinformationen. In unserer zweiten Studie befassen wir uns mit BDI, indem wir BWEs mit orthografischen Embeddings und einer Reihe anderer Merkmale anreichern und unser klassifikationsbasiertes System verwenden, um Schriftunterschiede zwischen Sprachen zu überwinden. Die dritte Veröffentlichung beschreibt besonders einfach zu extrahierende sprachübergreifende Signale, die bei der Entwicklung von Mapping-Ansätzen für BWEs berücksichtigt werden sollten, da sie einfach zugänglich sowie effektiv für das Bootstrapping des BWE-Mappins sind und das Versagen von unüberwachten Methoden überwinden. Der vierte Beitrag beschreibt unseren Ansatz zur Extraktion einer NE-Ressource für 1340 Sprachen, einschließlich sehr ressourcenarmer Sprachen aus allen wichtigen Bereichen der sprachlichen Vielfalt. Wir nutzen statistische Daten aus parallelen Korpora und Transliterationsmodelle, wodurch wir eine bessere Leistung als frühere Ansätze erzielen. Zuletzt modelliert unsere fünfte Studie die Annotationsprojektion als graphenbasiertes Labelpropagationsproblem für Part-of-Speech-Tagging (POS-Tagging). Part-of-Speech-Modelle, die auf unseren gelabelten Datensätzen trainiert wurden, übertreffen frühere Arbeiten für ressourcenarme Sprachen wie Bambara, Erzya, Manx und Yoruba.

Acknowledgments

This work would have never been possible without the support of many great people.

First of all, I would like to thank my supervisor, Hinrich Schütze, for his continuous support during my entire PhD. He guided me since the beginning and helped me finding my research direction.
Many thanks to Alexander Fraser and Viktor Hangya for their encouraging feedback and insights over these years. Thanks to François Yvon for the interesting discussions.
I would like to thank the reviewers, Danushka Bollegala and Pierre Zweigenbaum, for their availability and for taking the time to review this manuscript.
I would also like to thank everyone I met during my internships at Amazon. Their insightful discussions and support played a pivotal role in shaping my career path. Special thanks to Dennis Aumiller for being of invaluable help and support.
Thanks to all the amazing people I met at CIS. To my office mates Lütfi Kerem Şenel and Leonie Weissweiler, without whom coffee breaks would have never been that great. To Philipp Dufter, Ayyoob Imani, Masoud Jalili Sabet, Amir Hossein Kargaran, Abdullatif Köksal, and Peiqin Lin, for working on project together and for the brainstorming sessions. To all my colleagues for the enriching cultural discussions and lunches. To Thomas Schäfer for his technical support and Peggy Hobmaier for her assistance with paperwork.
Thanks to my Munich friends for all the "aperitivi", Bavarian trips, and for filling these years with countless cherished memories.

Thanks to Diego for always being by my side.
Finally, this work would have never been possible without the endless encouragement and support of my brother Lorenzo and my parents Antonella and Roberto.

Publications and Declaration of Co-Authorship

Chapter 2 corresponds to the following publication:

> **Silvia Severini***, Viktor Hangya*, Alexander Fraser, and Hinrich Schütze. "LMU bilingual dictionary induction system with word surface similarity scores for BUCC 2020.". In Proceedings of the 13th Workshop on Building and Using Comparable Corpora. 2020. *Equal contribution.

I conceived the original research contribution together with Viktor Hangya. I conceived the transliteration model and performed all the evaluations. Viktor Hangya ran the code to create Bilingual Word Embeddings. I regularly discussed the work with my advisor Hinrich Schütze and with Viktor Hangya. I wrote the initial draft of the article and did most of the subsequent corrections. All authors helped review the final draft of the paper and gave advice.

Chapter 3 corresponds to the following publication:

> **Silvia Severini**, Viktor Hangya, Alexander Fraser, and Hinrich Schütze. "Combining Word Embeddings with Bilingual Orthography Embeddings for Bilingual Dictionary Induction.". In Proceedings of the 28th International Conference on Computational Linguistics. 2020.

I conceived the original research contribution. I designed all the models and performed all the evaluations except for the VecMap-related evaluations (Viktor Hangya). I regularly discussed this work with my advisor Hinrich Schütze. I wrote the initial draft of the article and did most of the subsequent corrections. All authors helped review the final draft of the paper and gave advice.

Chapter 4 corresponds to the following publication:

> **Silvia Severini**, Viktor Hangya, Masoud Jalili Sabet, Alexander Fraser, and Hinrich Schütze. "Don't Forget Cheap Training Signals Before Building Unsupervised Bilingual Word Embeddings.". In Proceedings of the 15th Workshop on Building and Using Comparable Corpora. 2022.

I conceived of the original contribution of considering efficient and effective signals for Bilingual Dictionary Induction and exploiting romanization. I designed the algorithms and performed all the experiments. Viktor Hangya ran state-of-the-art models and contributed with regular discussions. I wrote the initial draft of the article and did most of the subsequent corrections. All authors helped review the final draft of the paper and gave advice.

Chapter 5 corresponds to the following publication:

> **Silvia Severini**, Ayyoob Imani, Philipp Dufter, and Hinrich Schütze. "Towards a Broad Coverage Named Entity Resource: A Data-Efficient Approach for Many Diverse Languages.". In Proceedings of the 13th Language Resources and Evaluation Conference. 2022.

I conceived of the original research contribution. I designed the method and performed all implementations and evaluations except for sections 5.2 and 5.3 conceived by me but implemented by Ayyoob Imani. I regularly discussed this work with my advisor Hinrich Schütze. I wrote the initial draft of the article and did most of the subsequent corrections. All authors helped review the final draft of the paper.

Chapter 6 corresponds to the following publication:

> Ayyoob Imani*, **Silvia Severini***, Masoud Jalili Sabet, François Yvon, and Hinrich Schütze. "Graph-Based Multilingual Label Propagation for Low-Resource Part-of-Speech Tagging.". In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. *Equal contribution.

Ayyoob Imani conceived the initial research contribution. I contributed with multiple ideas on the GNN model such as the integration of type-level information and the selection of training languages. I conceived the neural POS tagger and all the evaluations in the paper. I regularly discussed the work with my advisor Hinrich Schütze. Ayyoob Imani and I wrote the initial draft. All authors helped review the final draft of the paper and gave advice.

München, den 25. Oktober 2022

Silvia Severini

Contents

List of Figures

List of Tables

Acronyms

BERT	Bidirectional Encoder Representations from Transformers
BDI	Bilingual Dictionary Induction
BLI	Bilingual Lexicon Induction
BUCC	Building and Using Comparable Corpora
BWE	Bilingual Word Embeddings
CBOW	Continuous Bag-Of-Words
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit
GNN	Graph Neural Network
IR	Information Retrieval
LSTM	Long Short-Term Memory
mBERT	Multilingual BERT
MT	Machine Translation
NE	Named-entities
NER	Named-entity recognition
NLP	Natural Language Processing
OOV	Out-Of-Vocabulary
P@1	Precision at 1
POS	Part-of-Speech
RNN	Recurrent Neural Network
seq2seq	Sequence-to-Sequence
SVD	Singular Value Decomposition
XLM-R	XLM-RoBERTa, Unsupervised Cross-lingual Representation Learning at Scale
XNLI	Cross-Lingual Natural Language Inference

Chapter 1

Introduction

1.1 Motivation

The world is becoming more interconnected every day thanks to advances in transportation and technology. Consequently, the need to overcome language barriers is increasing. We are also experiencing an increase in language data collection from the billions of people who interact with web searches, social networks, emails, and customer services on a daily basis.

Natural Language Processing techniques contribute to the understanding of speech and text with computational models, which nowadays benefits from the vast amount of available data. Such models process text by splitting it into smaller units such as sentences, words, subwords, or characters which are encoded with real-value representations. With the advent of Statistical Natural Language Processing, rule-based NLP methods have been overtaken by Machine Learning approaches (Goldberg, 2017). The latter can exploit large amounts of textual data efficiently and contribute to the understanding of an increasingly large number of languages. However, there are more than seven thousand human languages on our planet (Eberhard et al., 2020), most of which are still not addressed by natural language technologies (Joshi et al., 2020; Blasi et al., 2022). Indeed, the basic resources for their processing, such as monolingual/bilingual corpora, dictionaries, and parsers, exist only for a small portion of them. Although languages are different under a myriad of aspects, such as origins, morphology, and phonology, cross-lingual commonalities can be extracted by analyzing multilingual corpora and linguistic constructions, with the goal of increasing the understanding of such lower-resourced languages.

One “linguistic bridge” between two or more languages is represented by words that have similar orthography. Such words are mostly Named Entities which are notably informative linguistic structures, often categorized as person, location, organization, and miscellaneous (Grishman and Sundheim, 1996). The similarity among entities in different languages can be evaluated with distance-based metrics at the character level on the original scripts, or on their romanized version. Moreover, named entities observed in parallel corpora often occur with similar frequency providing a further signal to interconnect different languages.

Embeddings encode language units with numerical vectors. One important feature is that they reflect the similarity between words by placing related words like "train", "bus", and "tram" close in their monolingual embedding space (Schütze, 1992; Mikolov et al., 2013b). Such monolingual spaces in two languages can be mapped onto each other to learn cross-lingual embeddings (Zou et al., 2013) which enable the learning of meaning across languages and build implicit links (Mikolov et al., 2013a; Artetxe et al., 2018; Lample et al., 2018b).

Finally, word aligners are another construct that plays an important role in language bridging (Brown et al., 1993; Och and Ney, 2000; Östling et al., 2016; Ngo-Ho and Yvon, 2019). Their goal is to find the source and target words in a pair of sentences that are translations of each other. Their application to multiparallel corpora allows for word-level linking of translated pairs.

The growing need for addressing under-resourced languages, to which not enough attention has been paid so far, together with the availability of cross-lingual links motivated us to investigate and build models that operate over languages with different scripts, word order, origins, and resource availability. In this thesis, we aim to tackle both semantic and syntactic tasks with character-level and word-level approaches that are mostly language-agnostic. In particular, we strive for languages for which resources are available in small quantities, or not at all, which are usually addressed as "low-resource" languages such as Yoruba (Niger-Congo), Manx (Indo-European), and Erzya (Uralic).

1.2 Approach

In this work, we approach the goal of multilinguality by exploiting multiparallel corpora and by investigating character-level information like the orthographic similarity between words in different languages. We aim to build language agnostic approaches that do not strictly rely on specific characteristics of a language such as word order, language family, or their scripts.

1.2.1 Research Questions

We aim to address the following research questions categorized into four groups:

(i) *Properties:* Which language links can be extracted in order to improve low-resource language understanding? Which properties do rare words in different languages have in common? How can orthography help to improve their translations?

(ii) *Extraction:* How can we extract named entity pairs for underresourced languages without language-specific signals? In particular, we investigate this for languages for which no named entity recognizer, annotated data, or pretrained language models are available.

(iii) *Models:* How can we model transliteration among different scripts? Are named entity pairs helpful to improve the performance of bilingual translations? How can they be incorporated?

(iv) *Applications:* How can we exploit parallel bilingual and multilingual corpora? Are multilingual/multiparallel corpora relevant for improving syntactic low-resource language understanding?

1.3 Dissertation Structure

In this chapter, we define the main topics that concern the publications in the rest of this thesis, conclusions, and future work. Chapters 2 and 3 describe our research on the general topic of building and using comparable corpora (it was published in the "Building and Using Comparable Corpora" workshop) and a follow-up paper which exploits orthography embeddings to improve Bilingual Dictionary Induction (BDI) performance focusing on low-frequency words. The following chapters focus not only on rare words, but also on low-resource languages. In particular, Chapter 4 shows our method to extract simple bilingual signals to tackle BDI and questions the strong trend towards unsupervised mapping approaches for Bilingual Word Embeddings construction. Chapter 5 describes our language-agnostic approach for building a multilingual named entity resource for 1340 languages, including very low-resource ones. Finally, Chapter 6 describes our formulation of annotation projection as a graph label propagation problem and shows its application to another sequence labeling task: Part-Of-Speech tagging.

1.4 Notation

Scalar values are lowercase italic letters (e.g., $t \in \mathbb{R}$), vectors are boldface lowercase letters (e.g., $\mathbf{v} \in \mathbb{R}^d$), and matrices are boldface uppercase letters (e.g., $\mathbf{W} \in \mathbb{R}^{d \times n}$). The i^{th} element of a vector $\mathbf{v}$ is addressed as v_i and the sequence of n elements in a vector is addressed as $(v_1, v_2, ..., v_n)$. $\mathbf{x}^T$ and $\mathbf{X}^T$ are the transposed vector and matrix. The inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted as $\mathbf{x}^T\mathbf{y}$ or

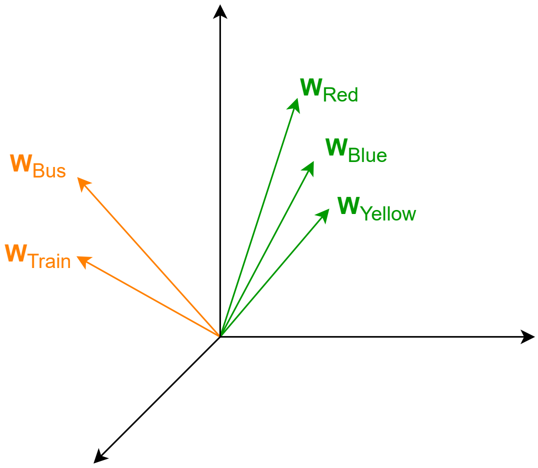


Figure 1.1: Schematic representation of word embedding vectors. Words that are semantically similar are encoded with similar vectors.

$\mathbf{x} \cdot \mathbf{y}$. The cardinality of a vector or set is denoted as $|\cdot|$. The cosine similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$ is calculated as $cos_{sim}(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v})/(||\mathbf{u}||\,||\mathbf{v}||)$ where $||\cdot||$ represents the Euclidean norm. Functions are represented with lowercase letters f. $f : A \rightarrow B$ denotes the feature and output space A and B of the function f. A tuple of a and b is represented as (a, b).

Throughout this work, we use ISO-639-3 codes (International Organization for Standardization) to uniquely identify each language with three letters (e.g., German-deu, Hindi-hin).

1.5 Word Representations

Word vectors, better known as word embeddings, represent words with numerical vectors derived from statistical or neural models (Turian et al., 2010). A schematic representation is depicted in Figure 1.1. Given a sentence $S = s_1, s_2, \ldots s_n$, where s_i is the i^{th} token in the sentence according to some tokenizer, $\mathbf{e}_i$ is the embedding of s_i in the embedding space E. Given a vocabulary V, which contains all the unique tokens (i.e., units) in the considered text, an embedding function is a mapping $e : V \rightarrow E$. In the case of the one-hot encoding (Hinton, 1984; LeCun et al., 2015), E is equal to $\{0, 1\}^{|V|}$ which assigns each i^{th} token in V the i^{th} unit vector in E. This creates representation vectors that are orthogonal to each other ($\forall \mathbf{e}_i, \mathbf{e}_j, i \neq j, \mathbf{e}_i^T \mathbf{e}_j = 0$), high-dimensional ($|V|$ is fixed and large), and sparse. Distributed representations instead use multiple elements to represent each vocabulary token (Deerwester et al., 1990; Schütze, 1992; Bengio et al., 2003;

Collobert et al., 2011; Mikolov et al., 2013b; Pennington et al., 2014; Baroni et al., 2014; Levy and Goldberg, 2014). E can therefore be a d-dimensional Euclidean space with $d << |V|$. A representation $\mathbf{e}_i$ is then a real-valued, dense, low-dimensional vector. Distributed embeddings also maintain the semantic information of a word by placing semantically similar words close to each other (i.e., often with high cosine similarity) and syntactic information (Andreas and Klein, 2014).

A distributed representation for s_i can be static, where $\mathbf{e}_i$ depends only on s_i, or contextualized, where $\mathbf{e}_i$ depends on more tokens in S. We now briefly describe both cases in monolingual and cross-lingual settings.

1.5.1 Static Embeddings

A static word embedding function maps each token to a vector:

$$e : V \rightarrow \mathbb{R}^d \tag{1.1}$$

with $d \in \mathbb{N}$. Such representations do not depend on their surrounding tokens. Since their introduction, different models have been proposed. Mikolov et al. (2013a,b) introduce the Word2Vec framework which can be modeled with one of the two neural network models: CBOW (Continuous Bag-of-Words) and Skip-gram. As depicted in Figure 1.2, CBOW aims at finding the unit w_t given a window of context units $w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}$, while Skip-gram aims at predicting the context units $w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}$ of a given unit w_t, that is whether each context word is likely to appear in the context window of w_t. GloVe (Global Vectors) from Pennington et al. (2014) represents another word embedding algorithm which combines global matrix factorization with a local context window. FastText (Bojanowski et al., 2017) is a fast and efficient method that is able to derive vectors also for out of vocabulary words, by taking morphological characteristics into account. FastText is of particular interest for this thesis, since it works well for creating rare word representations, as opposed to Word2Vec, so we elaborate on it in the following section.

1.5.1.1 FastText

FastText (Bojanowski et al., 2017) is derived from the Skip-gram model with negative sampling of Mikolov et al. (2013b). Given a training corpus of N words w_i with $i \in [1, N]$, the objective of the model is to maximize the following log-likelihood:

$$\sum_{n=1}^{N} \sum_{c \in C_n} \log p(w_c | w_n) \tag{1.2}$$

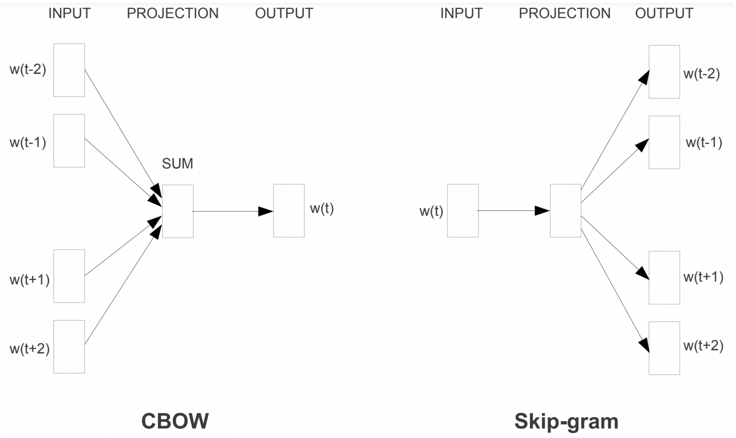


Figure 1.2: Representation of the CBOW and Skip-gram models. In CBOW, the representations of the context words are combined to predict the word in the middle, while Skip-gram does the opposite. The figure was taken from Mikolov et al. (2013a).

where C_n is the set of indices of words surrounding w_n. They also define a scoring function s which maps pairs of word w_n and context word w_c to a score in $\mathbb{R}$:

$$s(w_n, w_c) = \mathbf{u}_{w_n}^T \mathbf{v}_{w_c} \tag{1.3}$$

where $\mathbf{u}_{w_n}$ and $\mathbf{v}_{w_c}$ are the word vector and context word vector respectively.

For FastText, the scoring function is modified to capture the internal structure of words by representing a word by the sum of the vector representations of its *n*-grams. For example, given the word *"house"*, its *n*-grams with $n = 3$ are <*ho, hou, ous, use, se*> where $<$ and $>$ are boundary symbols included to distinguish prefixes and suffixes, in addition to the full word *"house"*. Given a dictionary of *n*-grams D, $D_{w_n} \subset 1, ..., |D|$ is the set of *n*-grams in w_n. Thus, the scoring function becomes:

$$s(w_n, w_c) = \sum_{d \in D_{w_n}} \mathbf{z}_d^T \mathbf{v}_{w_c} \tag{1.4}$$

where $\mathbf{z}_d$ is the vector representation of the *n*-gram d. In this way, the model captures information from the subwords of a word, allowing reliable embedding learning even for rare words.

We use these monolingual embeddings for building Bilingual Word Embeddings in Chapters 2, 3, and 4. In the next subsections, we describe contextualized embeddings which are currently used for a variety of classification and generation tasks. Note, however, that static embeddings like FastText are still crucial for the encoding of rare words and for cross-lingual tasks working with low-resource languages for which multilingual models are not available (see 1.5.3.2) or perform poorly (Muller et al., 2021).

1.5.2 Contextualized Embeddings

Static embeddings provide a single vector per word unit therefore failing to capture polysemy, since the meaning of a polysemous word depends on its context (Peters et al., 2018; Wang et al., 2020). For example, the word "tongue" in the sentences "Italian is my mother tongue" and "I accidentally bit my tongue" is represented in the same way even though it has two different meanings (language vs. the organ of taste). On the contrary, contextualized embeddings represent "tongue" differently according to its contexts.

A contextualized embedding function is defined as:

$$e : V^{t_{max}} \rightarrow \mathbb{R}^{t_{max}} \tag{1.5}$$

where t_{max} is the maximum number of tokens that the function can process at once (e.g., size of a phrase, sentence, paragraph, document).

McCann et al. (2017) first propose CoVe (Context Vector), a context vector model which uses a deep Long Short-Term Memory (LSTM) encoder to create contextual representations. These dynamic embeddings can be extracted from pretrained language models. Peters et al. (2018) introduce ELMo (Embeddings from Language Models) which extracts representations from the internal layers of a deep bidirectional language model pretrained on a large monolingual corpus. The forward language model computes the probability of a word w_i given the previous words $(w_1, w_2, \ldots, w_{i-1})$ in the sequence of words $(w_1, w_2, \ldots, w_N)$. Concurrently, Howard and Ruder (2018) propose ULMFit (Universal Language Model Fine-tuning) which is also based on a LSTM language model and on large unlabeled data. Both approaches are limited by the inability of LSTMs to capture long-range dependencies and by their computational inefficiency due to sequential computations.

Radford et al. (2018) use the Transformer model of Vaswani et al. (2017) together with positional embeddings to overcome these problems and introduce the GPT (Generative Pretraining) model. In order to allow models to work with both the left and right context of a word, Devlin et al. (2019) propose BERT (Bidirectional Encoder Representations from Transformers) which consists of a deep bidirectional Transformer model. BERT led to a number of other works introducing different pretraining objectives (Sun et al., 2019), token prediction techniques (Raffel et al., 2020; Lan et al., 2019), and efficiency strategies (Sanh et al., 2019; Liu et al., 2019).

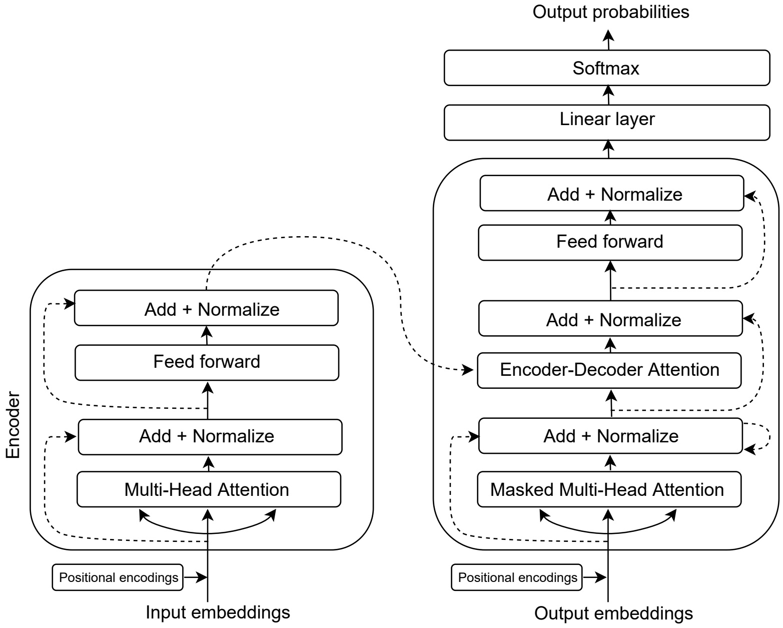


Figure 1.3: Schema of a Transformer model with one encoder and one decoder layer. Figure adapted from Alammar (2018) and Vaswani et al. (2017).

1.5.2.1 The Transformer Model

Previously, sequence-to-sequence problems were mainly addressed with RNNs (see 1.6.2). Attention (Luong et al., 2015b; Bahdanau et al., 2015) was introduced to mitigate the inefficiency of these architectures due to the processing of long sequences. However, the attention is computed recurrently as a weighted sum of all the past encoder states, meaning that both decoder and encoder at time t need to wait for the completion of $t - 1$ steps. This constraint leads to a time consuming and inefficient text processing when dealing with large corpora.

The Transformer architecture was introduced by Vaswani et al. (2017). It aims to address this recurrence problem by allowing for parallelization. A Transformer is made of stacked blocks. First, Positional Encodings (PE) are applied to the input embeddings simultaneously and the resulting vectors are passed to the first encoder block. Each block consists of a Multi-Head Self-Attention layer, a normalization layer which computes mean and variance across channels and spatial dimensions, residual skip connections around them, a feed-forward linear layer, and another normalization layer with residual connections. The decoder block is similar to the encoder one plus a Masked Multi-Head attention and an Encoder-Decoder attention which map the input sequence to the corresponding output word. In the original paper of Vaswani et al. (2017), the encoder and decoder are made of 6 blocks each. A schematic representation of this model is depicted in Figure 1.3. We now describe the main components separately.

Positional Encodings Since the Transformer computations can be done in parallel, it is necessary to inject positional information into the representations of each word. This is done via fixed *Position Encodings* (Gehring et al., 2017). In this way, if the same word appears at different positions, its representation will be slightly different depending on where it appears. These encodings are computed as follows:

$$\begin{aligned}\mathbf{PE}_{(pos,2i)} &= sin(\frac{pos}{10000^{2i/d_{model}}})\\ \mathbf{PE}_{(pos,2i+1)} &= cos(\frac{pos}{10000^{2i/d_{model}}})\end{aligned} \tag{1.6}$$

where $\mathbf{PE} \in \mathbb{R}^{T \times d_{model}}$, T is the maximum sequence length, d_{model} is the dimensionality of the embedding vectors, pos is the position in the sequence, and i is the hidden dimension index.

Attention Self-attention relates different positions of a sequence in order to compute its representation. For each token, the mechanism takes the encoder input and produces three vectors **q**, **k**, and **v**. Given an input sequence $s = (s_1, s_2, ..., s_n)$ with n tokens and their embeddings $\mathbf{X} \in \mathbb{R}^{N \times d_{model}}$, it computes:

$$\mathbf{K} = \mathbf{XW}^K, \mathbf{Q} = \mathbf{XW}^Q, \mathbf{V} = \mathbf{XW}^V \tag{1.7}$$

where $\mathbf{W}^K$, $\mathbf{W}^Q$, $\mathbf{W}^V$ are weight matrices. The self-attention of **X** is then computed as:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{QK}^T}{\sqrt{d_k}})\mathbf{V} \tag{1.8}$$

where the scaling factor d_k is the dimension of the keys. The result is a probability distribution determining the attention that should be given to each input token.

To gain further improvement and learn information from different representations, Vaswani et al. (2017) propose the *Multi-Head Attention*, which consists of m linear projections (heads) for **q**, **k**, and **v** to d_q, d_k, and d_v dimensions:

$$MultiHeadAttention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = concat(head_1, ..., head_m)\mathbf{W}^O \tag{1.9}$$

where $\mathbf{W}^O \in \mathbb{R}^{md_v \times d_{model}}$ is a projection matrix. Given $\mathbf{W}_i^Q$ and $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \times d_k}$, and $\mathbf{W}_i^V \in \mathbb{R}^{d_{model} \times d_v}$, each attention head is computed as:

$$head_i = Attention(\mathbf{QW}_i^Q, \mathbf{KW}_i^K, \mathbf{VW}_i^V) \tag{1.10}$$

In the decoder stage, the attention is limited to the previous word embeddings with respect to the current position:

$$MaskedAttention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T + \mathbf{M}}{\sqrt{d_k}})\mathbf{V} \tag{1.11}$$

where the mask **M** is a matrix made of zeros and $-\infty$.

1.5.3 Cross-lingual Embeddings

Previously described methods produce monolingual embedding, that is, only for a given language. We now describe cross-lingual embeddings, both static and contextualized, which encode words from two or more languages into a shared space still following the constraint that semantically similar words are closely located.

1.5.3.1 Static Multilingual Representation

Given two languages $l1$ and $l2$ with respective vocabularies V^{l1} and V^{l2}, their static embeddings are $\mathbf{E}^{l1}$ and $\mathbf{E}^{l2}$, which we call **F** and **G** for clarity. When the two embeddings are learned separately, they lie in two different spaces with no relationships. This means that the embeddings of “dog” in English and its translation “cane” in Italian are such that the cosine similarity of F_{dog} and G_{cane} is random instead of being close to 1 given their semantic link. Embeddings with such semantic relationships are referred to as Bilingual Word Embeddings (BWEs) if concerning two languages (Zou et al., 2013) or Cross-Lingual embeddings if concerning multiple ones. In order to create them, two main approaches have been proposed: *joint learning* and *mapping*.

Joint learning approaches aim to learn the embeddings F and G directly in a common embedding space. Different methods have been studied. Luong et al. (2015b) modify the Skip-gram model (Mikolov et al., 2013b) to jointly learn sentence and word-level alignments, and create BWEs for pairs of languages. Vulic and Moens (2015) propose to merge two aligned documents to create a pseudo-bilingual corpus and then train a Skip-gram model on it to create BWEs. Artetxe and Schwenk (2019) train a bidirectional LSTM on a large parallel corpus and jointly learn representations for 93 languages.

Mapping approaches are another way of creating BWEs. An illustrative example of English-Italian mapping is shown in Figure 1.4. Their goal is to learn **F** and **G** separately on large monolingual corpora, and subsequently learn a mapping function $w : \mathbb{R}^d \rightarrow \mathbb{R}^d$ to map one monolingual embedding space onto the other and create a cross-lingual space. The transformation is learned from word alignments or bilingual dictionaries (seed pairs) under the assumption

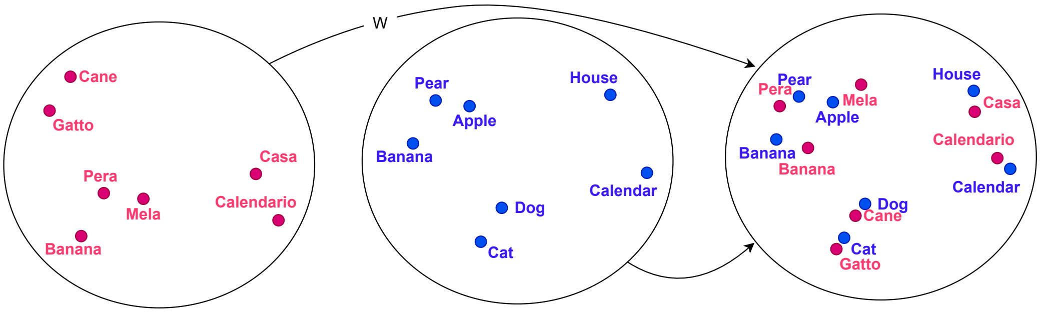


Figure 1.4: Example of mapping Italian monolingual embeddings (pink) onto English monolingual embeddings (blue) using a rotation matrix W.

that monolingual spaces are approximately isomorphic (Mikolov et al., 2013a). However, recent literature shows that the assumption does not hold for distant languages and in case of limited-size monolingual corpora (Søgaard et al., 2018; Ormazabal et al., 2019).

A similar formulation requires to find an orthogonal matrix $\hat{\mathbf{A}}$ that most closely maps the modified embedding matrices $\tilde{\mathbf{F}}$ to $\tilde{\mathbf{G}}$ in order to obtain embeddings in a shared space, where $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ consist only of embeddings from a bilingual dictionary D. The problem is known as the *Orthogonal Procrustes Problem* (Schönemann, 1966; Xing et al., 2015) where $\hat{\mathbf{A}}$ is orthonormal (i.e., $\hat{\mathbf{A}}^T\hat{\mathbf{A}} = \mathbf{I}_d$) and such that:

$$\hat{\mathbf{A}} = \operatorname*{argmin}_{A} ||\tilde{\mathbf{G}} - \tilde{\mathbf{F}}\mathbf{A}||_F^2 \tag{1.12}$$

where $||\cdot||_F$ represents the Frobenius norm. The solution can be found through Singular Value Decomposition (SVD): if $\tilde{\mathbf{G}}^T\tilde{\mathbf{F}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, then $\hat{\mathbf{A}} = \mathbf{V}\mathbf{U}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{\Sigma}$ is a diagonal matrix with non-negative singular values on its diagonal. However, this approach works under the assumption that the embedding matrices $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ only consist of embeddings from the given bilingual dictionary D.

Various methods have been proposed to create the initial dictionary. Supervised approaches can learn the mapping from 5000 seed word pairs (Mikolov et al., 2013a) down to only 25 pairs (Artetxe et al., 2017). Other approaches exploit weak signals like cognates (Smith et al., 2017), identical word pairs (Artetxe et al., 2017), or shared numerals (Søgaard et al., 2018) and achieve performance similar to that obtained with carefully created seed dictionaries. Recently, unsupervised methods to extract the initial seed lexicon or learn the initial mapping have been proposed (Lample et al., 2018a; Artetxe et al., 2018; Grave et al., 2019; Mohiuddin et al., 2020).

Among all the proposed mapping approaches, **VecMap** from Artetxe et al. (2018) represents a robust method that works generally well irrespective of the poor quality of the initial seed set. We use VecMap in Chapters 2, 3, and 4 to build BWEs. The method is based on the observation that two equivalent words in two languages, w^{l1} and w^{l2}, exhibit a close distribution of similarity values, given the similarity matrix of all words in the vocabulary.

Formally, given the word embedding matrices **F** and **G** in two different languages, VecMap aims to learn the linear transformation matrices $\mathbf{W}_F$ and $\mathbf{W}_G$ such that $\mathbf{FW}_F$ and $\mathbf{GW}_G$ are in the same cross-lingual embedding space. The method is made of four steps. First, the embeddings **F** and **G** are normalized. Second, VecMap performs a fully unsupervised initialization scheme that creates an initial seed set. To do so, it assumes that the embedding spaces **F** and **G** are isometric and thus the similarity matrices $\mathbf{M}_F = \mathbf{FF}^T$ and $\mathbf{M}_G = \mathbf{GG}^T$ are equivalent up to a permutation of their rows and columns. This permutation defines the dictionary across both languages. The third step is an iterative self-learning procedure that improves the initial dictionary **D** such that $D_{ij} = 1$ if the j^{th} word in **G** is the translation of the i^{th} word in **F**. It first learns the optimal mapping that maximizes the similarities of the initial dictionary and then computes a new optimal dictionary over the similarity matrix of the mapped embeddings. Formally, it computes:

$$\underset{\mathbf{W}_F, \mathbf{W}_G}{\operatorname{argmax}} \sum_i \sum_j D_{ij}((\mathbf{f}_{i*} * \mathbf{W}_F) \cdot (\mathbf{g}_{j*} * \mathbf{W}_G)) \tag{1.13}$$

where $\mathbf{f}_{i*}$ and $\mathbf{g}_{j*}$ denote the embeddings of the i^{th} and j^{th} words in their respective vocabularies and the similarity matrix over the mapping embeddings corresponds to:

$$\mathbf{FW}_F\mathbf{W}_G^T\mathbf{G}^T \tag{1.14}$$

The last step of VecMap is a final refinement through symmetric re-weighting, after the previous step has converged to a good solution.

VecMap suffers from the "Hubness problem" (Radovanovic et al., 2010) where a few points are nearest neighbors of many other points in high-dimensional spaces. Solutions have been proposed involving inverted nearest neighbor (Dinu et al., 2014), inverted Softmax (Smith et al., 2017), and Cross-domain Similarity Local Scaling (CSLS) (Lample et al., 2018a).

VecMap has both (semi-)supervised and unsupervised settings. The latter has been proven to work for a wide variety of language pairs, even when different families and scripts are involved. However, it fails for some very distant pairs like English/Chinese, English/Tamil, or English/Kannada leading to an accuracy score close to 0 for the Bilingual Dictionary Induction (BDI) task. Other more recent unsupervised approaches show similar behavior (Grave et al., 2019; Mohiuddin et al., 2020). In Chapter 4, we describe a simple technique to extract seed sets for such language pairs in an unsupervised way which leads to improved BDI scores. Our work is strong evidence that cheap bilingual signals for building BWEs should always be considered as baselines for unsupervised mapping approaches, especially if the two languages in question are very different (like English/Chinese and English/Tamil).

1.5.3.2 Contextualized Multilingual Representation

As mentioned in Section 1.5.2, one can extract contextualized embeddings from BERT. To obtain a multilingual version of them, Devlin et al. (2019) introduce *multilingual BERT* (mBERT) for 104 languages. It is pretrained on concatenated and shuffled data from Wikipedia with a shared vocabulary across all languages such that individual tokens can appear in multiple languages. Representations extracted from mBERT are effective for a variety of multilingual tasks including dependency parsing, cross-lingual Natural Language Inference (XNLI), and named entity recognition (NER). They were proved to be effective for zero-shot knowledge transfer (Pires et al., 2019), the setting in which the learner hasn't seen labeled examples of the language it is tested for.

Conneau and Lample (2019) propose XLMs (Cross-lingual Language Models) that use transformers and Translation Language Modeling to exploit parallel sentences with the goal of increasing the multilinguality of the models.

Conneau et al. (2020) introduce *XLM-RoBERTa* (XLM-R) which is based on a transformer masked language model trained with a larger shared vocabulary for 100 languages on CommonCrawl data for a total of 2.5TB. XLM-R outperforms mBERT on XNLI, cross-lingual Question Answering, and NER. We use XLM-R embeddings for the Part-Of-Speech tagging work that is the content of Chapter 6.

1.5.4 Character-level Embeddings

When dealing with text, a large vocabulary size to adequately represent the words or subwords in the corpus is necessary. Character-embeddings were proposed to improve model performance, as they include important language signals (Zhang and LeCun, 2015), and to limit the vocabulary size for efficiency reasons. Such embeddings are constructed similarly to word embeddings, but each vector represents a given character in a given language. Formally, given a word $c = c_1, c_2, \dots,$ c_n of length n, each character c_i is encoded with the respective embedding vector $\mathbf{v}_i$. For example, for the word "cat", there is a vector for each letter "c", "a", and "t". The resulting embedding of c can be obtained through Convolutional Neural Networks (CNN) (LeCun et al., 1998; O'Shea and Nash, 2015) or Recurrent Neural Networks (RNN) (Rumelhart et al., 1986; Werbos, 1990)(see 1.6.1 and 1.6.2) by looking at its character-level composition.

Despite their limited depth of semantic encoding compared to word embeddings, character-level embeddings are useful for a variety of tasks as they are able to

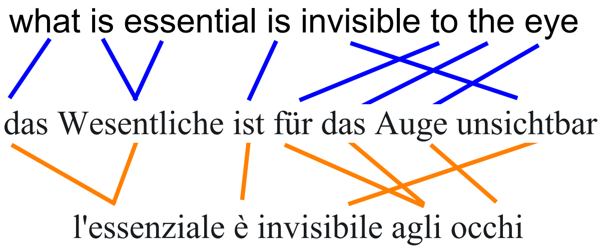

what is essential is invisible to the eye
das Wesentliche ist für das Auge unsichtbar
l'essenziale è invisibile agli occhi

Figure 1.5: Word alignment example for a quote from "The Little Prince" by Antoine De Saint-Exupéry (de Saint-Exupéry, 1943) in English, German, and Italian. Links between words indicate that the words are aligned (i.e., translations). Links between the English and Italian sentences are omitted for clarity of the drawing.

handle new, rare, and misspelled words and require a much smaller embedding matrix than for word-level embeddings. Examples are text classification, language modeling, and NER. In Chapter 3, we exploit character embeddings created with a sequence-to-sequence model to boost the performance for the Bilingual Dictionary Induction task.

1.5.5 Graph Neural Networks

We now describe Graph Neural Networks (GNNs), which create embeddings for graph-like structures. They constitute one of the building blocks of the work in Chapter 6 where they are used for POS tags projection, but can be used for projecting any classification labels between languages from multiparallel corpora.

In the real world, many objects are defined in terms of their connections to other objects such as images, social networks, and biological elements. They can be represented in the form of graphs. A graph $G = (N, E)$ represents the relations ($E = edges$) between a collection of elements ($N = nodes$). Edges can be *directed* if there are directional dependencies between nodes, or *undirected*. Graphs are usually represented with adjacency matrices of size $(n \times n)$ where n is the number of nodes in G. There are a variety of tasks when dealing with graph representations at the graph-level (predict properties of the entire graph), node-level (predict the role of each node), or edge-level (predict the relationships between nodes).

GNNs (Gori et al., 2005) have been introduced to leverage the structures and properties of graphs with neural networks. Unlike Convolutional Neural Networks, they can handle complex graph topologies. They use *node embeddings* that map a node to a d-dimensional space in which similar nodes are embedded close to each other. Given two nodes, u and v, we define an encoder function that converts the feature vectors $\mathbf{f}_u$ and $\mathbf{f}_v$ to $\mathbf{z}_u$ and $\mathbf{z}_v$ such that $sim(u, v) \approx \mathbf{z}_v^T \mathbf{z}_u$ where sim is a similarity function (e.g., based on Euclidean distance). The encoder function is obtained with neural networks and should be able to include local graph neighbors, aggregate information, and stack multiple layers. Given $\mathbf{h}_v^0 = \mathbf{x}_v$, a forward propagation is defined as:

$$\begin{aligned} a &= \mathbf{W}_k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} \\ b &= \mathbf{B}_k \mathbf{h}_v^{k-1} \\ \mathbf{h}_v^k &= \sigma(a+b) \quad foreach\ k = 1, \ldots, K \end{aligned} \tag{1.15}$$

where $\mathbf{x}_v$ is the feature vector of node v, $N(v)$ indicates the neighbors of node v, a represents the weighted average of all neighbors of node v, $\mathbf{B}_k$ is a weight bias matrix, and σ is the non-linear activation function. The final embedding of node v after K layers is $\mathbf{h}_v^K = \mathbf{z}_v$.

In Chapter 6, we exploit GNNs to propagate node labels from high to low-resource languages for the Part-Of-Speech tagging task (see 1.8.2). Given a sentence, words and a set of features are represented by nodes while alignment links between words in multiple languages are represented by edges, indicating translation relations. An example of the graph obtained with alignment links is shown in figure 1.5.

1.6 Character-level Models

NLP models usually work with words and subwords. Yet, processing text at the character-level has its own benefits and drawbacks. Such models generally cannot encode semantic information as word-level models do. However, in exchange, they drastically reduce the vocabulary size, which directly alleviates the computational burden on output predictions. Although the vocabulary size is smaller than for word-level models, it allows for the handling of spelling mistakes, abbreviations, and any arbitrary word, including rare words and jargon specific to a certain domain. Working at the character-level has also the advantage of relieving the need for a suitable tokenizer for the specific domain addressed. They have been employed for different tasks including sentiment analysis (Radford et al., 2017; Arora and Kansal, 2019), machine translation (Lee et al., 2017; Gao et al., 2020), and text classification (Zhang et al., 2015).

We describe below the two families of character-level models that are most commonly used: Convolutional and Recurrent Neural Networks based models. Note, however, that non-recurrent architectures, such as Transformers described in Section 1.5.2.1, have been recently investigated for character-level transduction tasks and adapted to favor models for specialized domains (El Boukkouri et al., 2020; Wu et al., 2021). They led to improved performance on morphological inflection, historical text normalization, grapheme-to-phoneme conversion, and transliteration.

1.6.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) (LeCun et al., 1998; O'Shea and Nash, 2015) have been introduced to deal with images and only later applied to text (Collobert et al., 2011). CNNs are based on the idea of sliding through text with a predefined window, called kernel or filter, in order to detect complex features of the data. They are made of two main layers: a convolutional layer to extract features and a pooling layer to reduce the size of the feature map.

Zhang et al. (2015) introduce character-level only CNNs for text classification.[1] The model takes a sequence of encoded characters with a vocabulary size of m and 1-of-m encoding (i.e., one-hot encoding). It is made of successions of convolutional and pooling layers and terminate with two fully-connected layers. They define a temporal convolutional module that computes 1-Dimensional convolutions to learn a diverse set of features. Formally, given $g : \{1, 2, \ldots, l\} \rightarrow \mathbb{R}$, a discrete input function, and $f : \{1, 2, \ldots, k\} \rightarrow \mathbb{R}$, a discrete kernel function, Zhang et al. (2015) define the convolution between $f(x)$ and $g(x)$ with stride s as:

$$h(y) = \sum_{x=1}^{k} f(x) \cdot g(y \cdot s - x + c) \tag{1.16}$$

where $h : \{1, 2, \ldots, \lfloor (l - k + 1)/s \rfloor\} \rightarrow \mathbb{R}$ and $c = k - s + 1$ is an offset constant. The model is then parameterized by a set of weights which are the kernel functions $f_{ij}(x)$ and a set of input $g_i(x)$ and output $h_j(y)$ features with $i \in \{1, 2, ..., m\}$ and $j \in \{1, 2, ..., n\}$. Therefore, h_j is defined as:

$$h_j(y) = \sum_{i=1}^{m} \sum_{x=1}^{k} f_{ij}(x) \cdot g_i(y \cdot s - x + c) \tag{1.17}$$

The model also contains 1-D temporal max-pooling modules that help training deep architecture by obtaining information while reducing the size of the feature vectors. Given the input function $g : \{1, 2, \ldots, l\} \rightarrow \mathbb{R}$, a max-pooling function $h : \{1, 2, \ldots, \lfloor (l - k + 1)/s \rfloor\} \rightarrow \mathbb{R}$ is defined as:

$$h(y) = \max_{x=1}^{k} g(y \cdot s - x + c) \tag{1.18}$$

[1] In the reminder of Section 1.6.1, we follow Zhang et al. (2015) notation.

1.6.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986; Werbos, 1990) have been widely used for word-level, but also character-level computations. A "Char-RNN" is a language model trained to predict the next character given a sequence of previous characters. Each character is encoded via 1-of-m encoding where m is the vocabulary size.

Given the input $\mathbf{x}^t$ at time step t, the current hidden state is calculated based on the current input at time step t and the previous time step's hidden state:

$$\mathbf{h}^t = f(\mathbf{U}\mathbf{x}^t + \mathbf{W}\mathbf{h}^{t-1} + \mathbf{b}) \tag{1.19}$$

where f is an activation function (e.g., *tanh*, *ReLU*), and $\mathbf{W}$ and $\mathbf{U}$ are weight matrices for the hidden-to-hidden and input-to-hidden connections respectively.

The RNN forward pass is then defined by 1.19 and the following set of equations

$$\begin{aligned} \mathbf{o}^t &= \mathbf{c} + \mathbf{V}\mathbf{h}^t \\ \hat{\mathbf{y}}^t &= softmax(\mathbf{o}^t) \end{aligned} \tag{1.20}$$

where V is a hidden-to-output weight matrix and $softmax(\mathbf{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$.

Sequence-to-sequence models (seq2seq) (Sutskever et al., 2014) are based on RNNs. In the next chapters, we show character-level seq2seq models for creating orthographic embeddings and learning transliterations. A seq2seq model takes a sequence of tokens as input $x = [x_1, x_2, ..., x_n]$ and outputs another sequence $y = [y_1, y_2, ..., y_m]$. It is composed of an *encoder* that captures x and a *decoder* that produces y. Both of them are usually made of LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) layers. Attention (Bahdanau et al., 2015; Luong et al., 2015a) plays an important role in the computation of long sequences in these models by allowing them to focus on different parts of x at every stage of y's generation. A schema of such models is shown in Figure 1.6. We now briefly describe LSTMs, GRUs, and Luong Attention.

1.6.2.1 Long Short-Term Memory

LSTMs were introduced by Hochreiter and Schmidhuber (1997) and are popular for a variety of tasks. Their main strength is the ability to handle long-range dependencies by avoiding the vanishing and exploding gradient problems of classical RNNs (Bengio et al., 1994; Khandelwal et al., 2018). The core idea is to use cell states made of three gates: input, forget, and output gates (i^t, f^t, o^t).

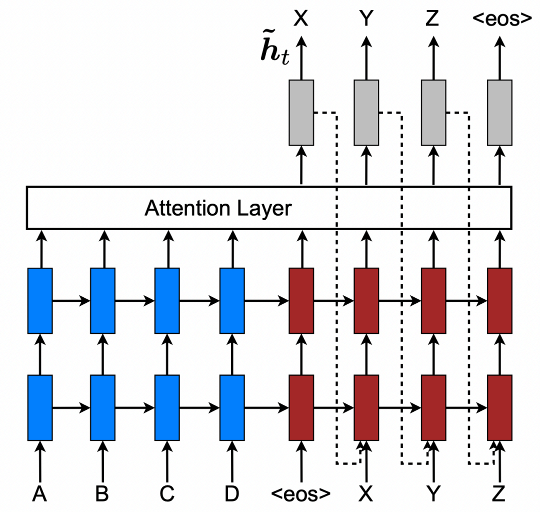


Figure 1.6: Schema of a Sequence-to-Sequence model with Attention for translating the input sequence [A,B,C,D] to the output [X,Y,Z]. Blue and red elements represents the *encoder* and *decoder* respectively. $\widetilde{h}_t$ represents the attentional vectors. Figure taken from Luong et al. (2015a).

LSTMs can be described with the following equations:

$$\begin{aligned}
\mathbf{i}^t &= \sigma(\mathbf{U}_i\mathbf{x}^t + \mathbf{W}_i\mathbf{h}^{t-1} + \mathbf{b}_i) \\
\mathbf{f}^t &= \sigma(\mathbf{U}_f\mathbf{x}^t + \mathbf{W}_f\mathbf{h}^{t-1} + \mathbf{b}_f) \\
\mathbf{o}^t &= \sigma(\mathbf{U}_o\mathbf{x}^t + \mathbf{W}_o\mathbf{h}^{t-1} + \mathbf{b}_o) \\
\mathbf{g}^t &= tanh(\mathbf{U}_g\mathbf{x}^t + \mathbf{W}_g\mathbf{h}^{t-1} + \mathbf{b}_g) \\
\mathbf{c}^t &= \mathbf{f}^t * \mathbf{c}^{t-1} + \mathbf{i}^t * \mathbf{g}^t \\
\mathbf{h}^t &= \mathbf{o}^t * tanh(\mathbf{c}^t)
\end{aligned} \tag{1.21}$$

where **U** and **W** are weight matrices, $\mathbf{c}^t$ is the cell state, $*$ represents element-wise multiplications, and σ is the sigmoid function. The input gates control how inputs contribute to the cell state, the forget gate decides on the previous cell state and the output gate controls which part of the state is propagated outside of the cell.

1.6.2.2 Gated Recurrent Unit

GRUs were introduced by Cho et al. (2014). They have only two gates, reset and update, which make them less complex and faster than LSTMs, but less accurate when dealing with longer sentences.

GRUs can be modeled with the following equations:

$$\begin{aligned}
\mathbf{r}^t &= \sigma(\mathbf{U}_r\mathbf{x}^t + \mathbf{W}_r\mathbf{h}^{t-1} + \mathbf{b}_r)\\
\mathbf{u}^t &= \sigma(\mathbf{U}_u x^t + \mathbf{W}_u\mathbf{h}^{t-1} + \mathbf{b}_u)\\
\mathbf{c}^t &= tanh(\mathbf{U}_c\mathbf{x}^t + \mathbf{W}_c(\mathbf{h}^{t-1} * \mathbf{r}^t) + \mathbf{b}_c)\\
\mathbf{h}^t &= (1 - \mathbf{u}^t) * \mathbf{c}^t + \mathbf{u}^t * \mathbf{h}^{t-1}
\end{aligned} \tag{1.22}$$

where $\mathbf{r}^t$ and $\mathbf{u}^t$ are the reset and update gates respectively, $\mathbf{U}$ and $\mathbf{W}$ are weight matrices, $*$ represents element-wise multiplications, and σ is the sigmoid function. When $\mathbf{r}^t$ is close to 0, the current hidden state ignores the previous one and keeps the current input, allowing to drop any previous irrelevant information. The update gates control the amount of information from the previous state that can be carried over, similar to the LSTM cell state.

1.6.2.3 Attention

We now describe the attention of Luong et al. (2015a) which we use in the following chapters. Bahdanau et al. (2015) define it similarly, with differences in the simplification of the original model. Luong's attention can be divided into two categories based on whether the attention is placed on all input positions or only on a few ones: *global* and *local*. In both cases, given an input sequence $x_1, \ldots, x_T$, the goal is to compute each context vector $\mathbf{c}_t$ which captures the source information needed to predict each target word y_t.

The context vector is calculated as:

$$\mathbf{c}_t = \sum_{i=1}^{T} \alpha_{t,i}\mathbf{h}_i \tag{1.23}$$

where $\mathbf{h}_i$ corresponds to the i^{th} hidden state of the encoder and $\alpha_{t,i}$ are weight values obtained as:

$$\alpha_{t,i} = softmax(a(\mathbf{s}_t, \mathbf{h}_i)) \tag{1.24}$$

where $a(\cdot)$ is an alignment model that uses the annotations and the current decoder hidden state $\mathbf{s}_t$.

For the *global* attention, $a(\cdot)$ can be calculated with one of the following options:

$$a(\mathbf{s}_t, \mathbf{h}_i) \begin{cases} \mathbf{s}_t^T\mathbf{h}_i & Dot\\ \mathbf{s}_t^T\mathbf{W}_a\mathbf{h}_i & General\\ \mathbf{v}_a^T tanh(\mathbf{W}_a[\mathbf{s}_t; \mathbf{h}_i]) & Concat \end{cases} \tag{1.25}$$

where $\mathbf{W}_a$ is a trainable weight matrix and $\mathbf{v}_a$ is a weight vector.

The *local* attention, instead, is applied to a subset of the source words to address the limitation of the global attention when working with longer sequences. The context vector $\mathbf{c}_t$ is computed as a weighted average of $\mathbf{h}_i$ within a window centered over p_t with D chosen empirically: $[p_t - D, p_t + D]$. There are two variants to define p_t: monotonic alignment where $p_t = t$ and predictive alignment where p_t is predicted by the model.

1.7 Transliteration

Transliteration is the process of converting text from a source language to a target language. Alice/أليس (alys) and London/Лондон (London) are examples of English/Arabic and English/Russian transliteration pairs. Formally, a transliteration model takes a source word s as input and outputs one or more target pairs $(t_1, p_1), ..., (t_n, p_n)$ sorted by p_i such that (t_i, p_i) is a tuple where t_i is the i^{th} transliteration of s with a probability p_i (Karimi et al., 2011). A transformation rule to transform s to t is denoted as $s \rightarrow (t_i, p_i)$ where s is in the source language alphabet, t_i is a (sub)string in the target alphabet, and p_i is the probability of transliterating s to t.

Transliteration concerns the following linguistic concepts:

- *Phoneme*: it is the smallest unit of speech that distinguishes meaning. Methods use phonemes to break down words before transliterating them (Oh and Choi, 2002; Virga and Khudanpur, 2003). The intuition behind it is that phonetical representations are shared across languages so it can be used as an intermediate form between two languages.

- *Grapheme*: it is the fundamental unit in written languages and includes alphabetic letters, numerals, punctuation marks, and all the individual symbols in a given language. Multiple graphemes may represent a single phoneme. For example, the word “shine” contains 5 graphemes but only 3 phonemes since “sh” and “ne” corresponds to just one phoneme each.

- *Writing system*: it is a symbolic system that represents expressible elements in a language. It can be categorized as *logographic* (e.g., Chinese), *syllabic* (e.g., Japanese), *featural* (e.g., Korean), and *alphabetic* or *segmental* (e.g., Arabic and Latin systems).

Romanization is a type of transliteration which consists of transliterating words into Latin characters. In Chapter 4 we exploit the Uroman romanization tool (Hermjakob et al., 2018). It performs a less accurate transliteration than when using transliteration models since it uses 1-1 character correspondences and it doesn’t vowelize text that lacks explicit vowels as for Arabic and Hebrew. However, it is applicable to many languages and it is of sufficient quality for our approaches.

1.7.1 Challenges

Building transliteration methods brings with it a number of challenges (Karimi et al., 2011).

First, each language has its own sounds. This implies that a sound may be missing from one language to another. Systems need to learn the convention of writing the missing sounds which are substituted with sequences of sound units and letters in the target language.

Second, a word may be transliterated in different ways, according to different dialects of the human transliterators or to the novelty of the word which makes the standardization of transliterations difficult. For example, the Hindi word गुलाब (rose) can be transliterated as "gulaab", "goolaab", or "gulab" (Prabhakar and Pal, 2018).

Third, deciding when a word should be transliterated or translated remains a difficult challenge. For example, given the named entity "Lake Garda", only the second word should be transliterated since the first word is an ordinary word of English. For example, the Russian pair would be " Озеро Гарда " (Ozero Garda). Hermjakob et al. (2008) propose to include a transliteration component to machine translation systems to mitigate this problem. In Chapter 3, we propose to integrate our Bilingual Orthography Embeddings, created with a seq2seq transliteration model, into a classification approach with the goal of correctly understanding when to translate or transliterate words.

Lastly, many systems rely on transliteration tables (i.e., character correspondences for language pairs) (Stalls and Knight, 1998) or labeled data (Rosca and Breuel, 2016; Shao and Nivre, 2016). However, such resources are available in small quantities or not at all for low-resource languages which highlights the relevance of unsupervised transliteration approaches that exploit language links. In Chapter 2 and 3 we learn transliteration models from unlabeled data (i.e., no explicit transliteration information) and in Chapter 5 we exploit similarity measures and co-occurrence statistics from unlabeled parallel corpora.

1.7.2 Approaches

Karimi et al. (2011) and Prabhakar and Pal (2018) provide comprehensive surveys on transliteration approaches. More generally, this task can be categorized into two subtasks: generation and mining.

Transliteration *generation* aims to produce the correct transliteration in the target language for a given source language word. Given two languages, approaches

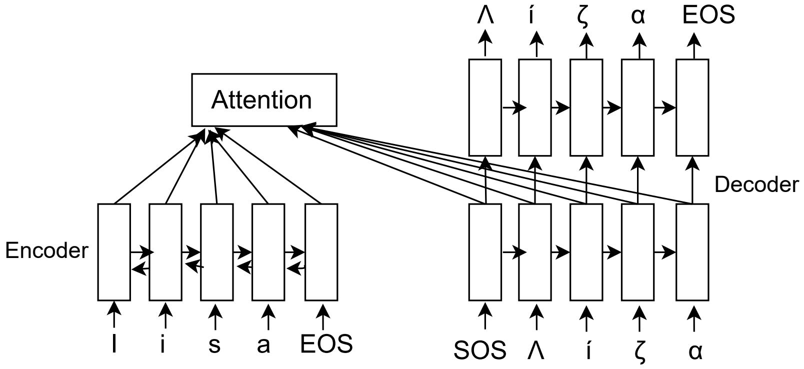


Figure 1.7: Representation of a character-level sequence-to-sequence model with attention for neural machine transliteration representing the transliteration of the English name "lisa" to Greek.

can be based on their phonetic similarity (Knight and Graehl, 1998; Oh and Choi, 2002), orthographic similarity (Li et al., 2004; Wang et al., 2015), or a combination of the two (Oh and Choi, 2006; Finch et al., 2012).

Transliteration *mining* aims at finding word pairs in comparable or parallel corpora, or in the Web. This can be used for building bilingual lexicons as for our named entity resource in Chapter 5. Similar to generation, mining is performed using phonetic similarity measures (Jiampojamarn et al., 2010; Dasgupta et al., 2013), word co-occurrence information (Wu et al., 2012), and statistical models (Sajjad et al., 2012).

Neural machine transliteration emerged as part of machine translation to deal with proper nouns and terms that are translated preserving their sound or pronunciation. To this end, the encoder-decoder architecture of Sutskever et al. (2014) has become a good alternative to other statistical methods. Rosca and Breuel (2016) propose to use such an architecture, in particular, attentional sequence-to-sequence models (seq2seq) to learn transliterations at the character level (see Section 1.6). A schematic representation of a seq2seq model for transliteration is shown in Figure 1.7. Wu and Yarowsky (2018) compare transliteration performance for different methods for translations such as phrase-based statistical MT, grapheme-to-phoneme systems, and seq2seq. More recently, studies have analyzed the application of the Transformer architecture to this task (Moran and Lignos, 2020; Wu et al., 2021). They show that transformer-based architectures outperform seq2seq models with careful hyperparameter tuning, especially when insertions and substitutions are involved for the many-to-one multilingual paradigm (i.e., multiple languages to English).

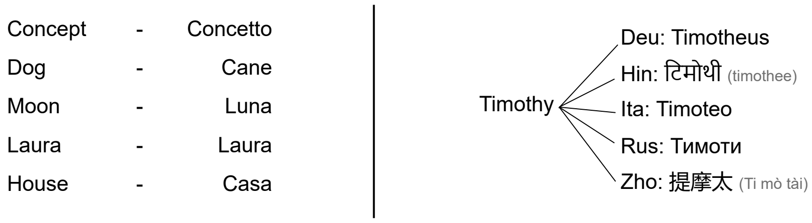


Figure 1.8: Examples of word pairs for the Bilingual Dictionary Induction task English/Italian (left) and cross-lingual Named Entities Extraction (right).

1.7.3 Applications

Transliteration is a crucial component for removing language and scriptural barriers. Different NLP tasks benefit from it such as Cross-lingual Information Retrieval and Machine Translation.

A significant portion of tokens that are not covered by lexicons are proper nouns, domain-specific terms (Meng et al., 2004), or foreign words leading to the so-called “Out-Of-Vocabulary” (OOV) problem. This lack of coverage affects the performance of cross-lingual retrieval tasks where these terms carry specific and relevant information to understand the text. However, these words, especially proper nouns, are often transliterated. Following this intuition, systems have been equipped with transliteration knowledge (Virga and Khudanpur, 2003).

In Machine Translation, transliteration components can be used to distinguish between translated and transliterated words (Hermjakob et al., 2008) or to mitigate the OOV problem by integrating a transliteration system into MT models (Finch et al., 2011; Durrani et al., 2014).

Following the intuition that Named-Entities (NE) are often transliterated, transliteration models can also be exploited for NE generation and mining (Moran and Lignos, 2020). In Chapter 5, we use a transliteration model to create a NE resource for more than a thousand languages.

1.8 Evaluation Tasks

The methods and models proposed in the remainder of this thesis have been evaluated on four main tasks described in this section: Bilingual Dictionary Induction, Part-of-Speech tagging, Annotation Projection, and Named Entities Extraction. We describe them from a cross-lingual perspective since they have been used with the goal of evaluating our multilingual methods.

1.8.1 Bilingual Dictionary Induction

Bilingual Dictionary Induction (BDI), also known as Bilingual Lexicon Induction, is the task of inducing word translations in two languages from monolingual corpora. These corpora can be completely unrelated or comparable (i.e., containing related information) (Rapp, 1995; Chiao and Zweigenbaum, 2002; Sharoff et al., 2013). The words for which we want to find a translation are referred to as *source words*, to be mapped to *target words*. An example is shown in Figure 1.8. Getting these pairs is potentially useful for low-resource Machine Translation when only small bilingual corpora are available. In particular, the mining of out-of-vocabulary words (OOV) is crucial to producing accurate translations.

BDI is commonly used to evaluate Bilingual Word Embeddings (see 1.5.3.1) by calculating the cosine similarity of the word pairs in two different languages and taking the n closest target words as translation candidates for each source word. However, BWE-based methods perform poorly for rare words due to their poor embedding representation (Braune et al., 2018; Czarnowska et al., 2019). In Chapters 2 and 3 we propose to improve the performance of BDI on low-frequency words by integrating orthography information. We exploit the normalized Levenshtein distance (Levenshtein et al., 1966) and seq2seq transliteration models to allow the processing of languages with different scripts.

BDI performance can be evaluated with accuracy at k, which considers the top-k predictions, or with F1 scores. A longterm effort to standardize the evaluation of this task is the "Building and Using Comparable Corpora" (BUCC) shared task (Sharoff et al., 2015), which aims at mining multilingual lexical knowledge from comparable corpora. We submitted our work on Chapter 2 to the 4th edition of the BUCC shared task (Rapp et al., 2020) and achieved the best results for Russian.

1.8.2 Part of Speech Tagging

Syntax defines the grammatical structure of a sentence. Part-Of-Speech tagging, syntactic analysis, and dependency parsing are examples of NLP tasks that involve syntax. We now describe the first one, as it is relevant to our work in Chapter 6.

Part-of-speech (POS) tagging consists of automatically assigning the syntactic annotations to words. Examples of POS tags are adverbs (ADV), proper nouns (PROPN), or prepositions (PREP). For instance, the sentence "I like to read books" gets the following POS tags:

I	like	to	read	books
PRON	*VERB*	*PART*	*VERB*	*NOUN*

It is a sequence labeling task which is a special type of classification aiming to predict the label sequence $y = (y_1, ..., y_N)$ for a given sequence of tokens

$s = (s_1, ..., s_N)$ represented by feature vectors $(\mathbf{f}_1, ..., \mathbf{f}_N)$. Ways to address this task include rule-based tagging and a probabilistic method that tags a word based on its frequency on an already tagged corpus. A simplistic approach is to solve:

$$\hat{y}_i = \underset{y_i}{\operatorname{argmax}}\, P(y_i|\mathbf{f}_i) \quad for\, each \;\; i = 1, \ldots, N \tag{1.26}$$

However, this formulation does not consider any sequential patterns in the data like neighborhood information (i.e., context). To do so, *Markov models* (Gao and Johnson, 2007) and *Conditional Random Fields* are applied (Awasthi et al., 2006). More recently, Neural Network methods trained on labelled data took the lead on this task achieving high accuracy scores either on a single language or in a multilingual setting (e.g., Akbik et al. (2018); Heinzerling and Strube (2019); Kondratyuk and Straka (2019); Akhil et al. (2020); Besharati et al. (2021)). This labeled data used for models training is scarce or completely unavailable for low-resource languages. To create such training data, in Chapter 6 we propose our method to annotate POS data by annotation projection (see 1.8.3) from more high-resource languages.

1.8.3 Annotation Projection

Annotation projection is the task of transferring labels from one language to another. Given a parallel corpus X in the source language and a corpus Y in the target language, the goal is to project labels from X to Y using cross-lingual links (e.g., word alignment information). It is particularly useful when the target language has zero or few labeled examples to enable the training of supervised neural models that can then be applied to the target language.

The approach was introduced by Yarowsky and Ngai (2001) who project POS labels and noun-phrase structures across languages using parallel corpora and word aligners. Since then, the approach has been applied also to NER (Yarowsky et al., 2001), word sense tagging (Bentivogli et al., 2004), semantic role labeling (Padó and Lapata, 2005), and dependency parsing (Hwa et al., 2005). To further boost transfer performance, researchers experiment with multi-source language projection (Fossum and Abney, 2005; Agić et al., 2016; Eskander et al., 2020) exploiting massively multi-parallel resources to make up for noisy and limited data. Other approaches use auxiliary lexicons as additional training signals to guide the learning (Täckström et al., 2013; Plank and Agić, 2018). In Chapter 6, we propose to project tags by formulating the problem as a multi-parallel graph projection without using annotated data for the target languages.

1.8.3.1 Multi-parallel corpora

Multi-parallel corpora contain aligned sentences in more than two languages. Examples of such corpora are the Parallel Bible Corpus of Mayer and Cysouw (2014) which we extensively use in the following chapters, the Proceedings of the European Parliament (Koehn, 2005), JW300 from Agić and Vulić (2019), and Tatoeba from Tiedemann (2020). Although their size is generally smaller than bilingual corpora, they support many low-resource languages not usually covered by language resources and are therefore crucial for their analysis. Ultimately, they can be used as language bridges to improve, among others, machine translation (Cohn and Lapata, 2007), embedding learning (Dufter et al., 2018), word alignment (Imani et al., 2022), and annotation projection (Agić et al., 2015).

1.8.4 Named Entities Extraction

Named entities (NE) are real-world objects that are referred to with proper nouns such as people, locations, and organizations; examples are *Charles Darwin*, *Munich*, and *Amazon*. They are crucial for many NLP tasks such as information retrieval, question answering, and entity linking.

The goal of cross-lingual Named Entities extraction is to build bilingual or multilingual resources with names translated in two or more languages by finding them in unstructured texts, usually parallel corpora. An example for the English name Timothy is shown in Figure 1.8. Note that this task differs from Named Entity Recognition since the latter aims to find named entities in text and classifying them with predefined labels such as person, location, organization, miscellaneous, and many more categories (Mai et al., 2018). The NE extraction task can be solved using parallel corpora together with named entity recognizers (Li et al., 2020) or word aligners (Wu et al., 2018), using large monolingual corpora from which one can extract high-quality static or contextualized embeddings (Wu et al., 2020; Li et al., 2021), with gazetteers (Torisawa et al., 2008), or via Wikipedia hyperlinks (Tsai and Roth, 2016). However, these elements are scarce or not available for very low-resource languages such as Kannada, Georgian, or Bambara. To make up for these issues and create an NE resource for a variety of languages, we propose our extraction method in Chapter 5 based on co-occurrence statistics and transliteration.

1.9 Conclusion

This chapter introduced the topics relevant to the thesis and to the next chapters. We described monolingual and multilingual word representations, basic character-level and syntax-level models, the transliteration paradigm, and the evaluation tasks that are encountered in the rest of this work.

1.9.1 Contribution

Considering the research questions formulated in 1.2.1, we can summarize our contribution as follows:

(i) *Properties:* We identified the concept of transliteration as a crucial signal to interconnect different languages. Transliteration pairs are usually proper nouns and domain specific terms which are often considered rare words in monolingual corpora. We used them to bootstrap mapping approaches (Chapter 4) and improve Bilingual Dictionary Induction performance on low-resource languages.

(ii) *Extraction:* We proposed to exploit co-occurrence statistics of named entities (NE) in parallel corpora to bootstrap neural transliteration models in Chapter 5. By applying our language-agnostic approach to the Parallel Bible Corpus of Mayer and Cysouw (2014), we created a NE resource for 1340 languages which represents one of the first published cross-lingual lexicons for some of the addressed low-resource languages.

(iii) *Models:* We proposed to exploit orthographic similarities among NE pairs as additional information for creating bilingual dictionaries. By using transliteration models, we propose methods that also work for languages with different scripts for which the edit-distance similarity metric wouldn't help (Chapter 2). We proposed Bilingual Orthography Embeddings to model orthographic similarities between words in different languages and proposed to combine them to BWEs to improve Bilingual Dictionary Induction performance for such low-frequency words (Chapter 3).

(iv) *Applications:* We proposed to exploit multiparallel corpora to formulate the task of word annotation projection as a graph label propagation problem. We applied our method to the Part-Of-Speech tagging task by transferring labels from high-resource to low-resource languages. Using our POS models trained on the projected data, we improved the state-of-the-art for low-resource languages (Chapter 6).

1.10 Future work

We now describe possible future work based on our research questions:

(i) *Properties:* The projection method of Chapter 6 is conducted from high-resource languages to low-resource ones. However, studies indicate that the knowledge transfer can be improved when it happens among languages with similar origins; writing systems, word orders, and lexical-phonetic distance significantly impact cross-lingual performance (Eskander et al., 2020; Zhou and Waibel, 2021; de Vries et al., 2022). It is still unclear which factor is the predominant one; this makes it interesting to investigate which characteristics and properties may favor the transfer in our graph-based modeling.

(ii) *Extraction:* Multiparallel corpora can be used to build alignment graphs as we showed in Chapter 6. However, they provide important information to link languages and should be considered for further extractions. For example, they could be exploited to mine seed pairs through word alignment links and serve as anchor points for aligning monolingual embeddings (Eder et al., 2021) and building bilingual ones. If links between more than two languages are available, one may create a chain of embedding mappings in order to build multilingual ones.

(iii) *Models:* To the best of our knowledge, our NEs resource is one of the few for many low-resource languages (Chapter 5). We have already shown its usefulness for cross-lingual mapping of word embeddings for a few of them. However, it would be interesting to use these NEs as seed signals to bootstrap other tasks, such as mapping methods (e.g., VecMap), for all the 1340 languages available. Moreover, the names in our resource were not post-processed to remove prefixes and suffixes making them appealing for morphological studies of the language (e.g., the meaning of a specific affix).

(iv) *Applications:* In Chapter 6, we describe our approach to label propagation for POS tagging and show that it performs well for low-resource languages. The POS task is only one sequence tagging task in which each word gets assigned a label. Another example is the Named-Entity Recognition (NER) task where each word is categorized with an entity tag to indicate the type of element such as person, location, and organization, or a null tag. Given the similarity of the two tasks, an interesting application would be to apply our method to NER. We have already conducted preliminary studies in this direction and saw that NER drastically suffers from the domain shift problem caused by the different domains of training and test corpora, biblical versus generic, that we used in our POS work which makes it more challenging.

Finally, our goal of multilinguality can be pushed forward by applying our NEs extraction method of Chapter 5 to more languages since there exist more than 7000 languages in the world and we were able to address only around 20% of them. This would require the non-trivial matter of getting access to additional parallel corpora for the missing languages which do not necessarily need to be as multiparallel as PBC, but can simply be bilingual (e.g., English/Language) augmenting the probability of finding them.

Chapter 2

Corresponds to the following publication:

> **Silvia Severini***, Viktor Hangya*, Alexander Fraser, and Hinrich Schütze. "LMU bilingual dictionary induction system with word surface similarity scores for BUCC 2020.". In Proceedings of the 13th Workshop on Building and Using Comparable Corpora. 2020. *Equal contribution.

Declaration of Co-Authorship: I conceived the original research contribution together with Viktor Hangya. I conceived the transliteration model and performed all the evaluations. Viktor Hangya ran the code to create Bilingual Word Embeddings. I regularly discussed the work with my advisor Hinrich Schütze and with Viktor Hangya. I wrote the initial draft of the article and did most of the subsequent corrections. All authors helped review the final draft of the paper and gave advice.

Proceedings of the 13th Workshop on Building and Using Comparable Corpora, pages 49–55
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020
© European Language Resources Association (ELRA), licensed under CC-BY-NC

LMU Bilingual Dictionary Induction System with Word Surface Similarity Scores for BUCC 2020

Silvia Severini*, Viktor Hangya*, Alexander Fraser, Hinrich Schütze
Center for Information and Language Processing
LMU Munich, Germany
{silvia, hangyav, fraser}@cis.uni-muenchen.de

Abstract

The task of Bilingual Dictionary Induction (BDI) consists of generating translations for source language words which is important in the framework of machine translation (MT). The aim of the BUCC 2020 shared task is to perform BDI on various language pairs using comparable corpora. In this paper, we present our approach to the task of English-German and English-Russian language pairs. Our system relies on Bilingual Word Embeddings (BWEs) which are often used for BDI when only a small seed lexicon is available making them particularly effective in a low-resource setting. On the other hand, they perform well on high frequency words only. In order to improve the performance on rare words as well, we combine BWE based word similarity with word surface similarity methods, such as orthography and transliteration information. In addition to the often used top-n translation method, we experiment with a margin based approach aiming for dynamic number of translations for each source word. We participate in both the open and closed tracks of the shared task and we show improved results of our method compared to simple vector similarity based approaches. Our system was ranked in the top-3 teams and achieved the best results for English-Russian.

Keywords: BDI, BWE, Orthography, Transliteration

1. Introduction

Bilingual Dictionary Induction is the task of inducing word translations from monolingual corpora in different languages. It has been studied extensively as it is one of the main tasks used for evaluating the quality of BWE models (Mikolov et al., 2013b; Vulic and Korhonen, 2016). It is also important for downstream tasks such as translating out-of-vocabulary words in MT (Huck et al., 2019).

Although there is a large amount of work for BDI, there is no standard way to measure the performance of the systems, the published results are not comparable and the pros and cons of the various approaches are not clear. The aim of the BUCC 2020 – *Bilingual Dictionary Induction from Comparable Corpora* – shared task (Rapp et al., 2020) is to solve this problem and compare various systems on a standard test set. It involves multiple language pairs including Chinese, English, French, German, Russian and Spanish and supports comparable monolingual corpora, and training and testing dictionaries for high, middle and low frequency words. In this paper, we present our approach to the shared task and show results on English-German and English-Russian.

BWEs are popular for solving BDI by calculating cosine similarity of word pairs and taking the n most similar candidates as translations for a given source word. They were shown to be very effective for the task using a small seed lexicon only (e.g., (Mikolov et al., 2013b)) as opposed to MT based approaches where parallel data is necessary. In addition, Conneau et al. (2018) and Artetxe et al. (2018) were able to learn BWEs without any seed dictionaries using a self-learning method that starts from an initial weak solution and improves the mapping iteratively. Due to this, BDI is one of the building blocks of unsupervised MT and are particularly relevant in low-resource settings (Artetxe et al., 2019; Lample et al., 2018).

Although BWE based methods work well for translating high frequency words, it was shown that they tend to have low performance when translating low-frequency words or named entities due to poor vector representation of such words (Braune et al., 2018; Riley and Gildea, 2018; Czarnowska et al., 2019). By using character n-gram representations and Levenshtein similarity of words, Braune et al. (2018) showed improved results on rare and domain specific words. Similarly, Riley and Gildea (2018) improves the translation of such words by integrating orthographic information into the vector representation of words and in the mapping procedure of BWEs. On the other hand, these techniques are only applicable in the case of language pairs having the same scripts. Recently, Riley and Gildea (2020) proposed an unsupervised system based on expectation maximization and character-level RNN models to learn transliteration based similarity, i.e., edit distance similarity across different character sets. To train their system they took $5,000$ word pairs having the highest cosine similarity based on BWEs. However, this method could be noisy, since non-transliteration pairs could be generated as well.

In this paper, we present our approach to BDI focusing on the problems of low frequency words translation. We follow the approach of Braune et al. (2018) and improve low frequency translation by combining a BWE based model with other information coming from word surface similarity: orthography and transliteration. The orthographic model is used in the case of word pairs with shared alphabet and uses the Levenshtein similarity. The transliteration model is used for pairs with different scripts where an orthographic comparison would not be possible and it is obtained from our novel fully unsupervised transliteration model. In contrast to (Riley and Gildea, 2020), we propose a cleaning method for filtering non-transliteration pairs from the used dictionary before training the model to ensure a less noisy training

[*] The authors contributed equally to this manuscript.

signal.

We test our system on the *English-German* pairs (En-De, De-En) and *English-Russian* pairs (En-Ru, Ru-En) provided in the BUCC 2020 Shared Task (Rapp et al., 2020). We participate in both the open and closed tracks of the shared tasks, using embeddings extracted either from *Wikipedia* (Conneau et al., 2018) or *WaCKy* (Baroni et al., 2009) respectively. In addition to using a static number of most similar words as translation, we experimented with methods returning a dynamic number of translations given each source word.

In the rest of the paper, we first describe the approach and how we obtain the two word surface similarity scores. Then, we present the experiments on the BUCC 2020 dataset and discuss the results.

2. BUCC 2020 Shared Task

The BUCC 2020 Shared Task (Rapp et al., 2020) focuses on multilingual lexical knowledge extraction from comparable rather than from parallel corpora. It gives the opportunity to experiment with the BLI task providing corpora and bilingual datasets for different language pairs. It also provides training data and a common evaluation framework.

The shared task is divided into open and closed tracks. In the open track participants are allowed to use their own corpora and training data, whereas in the closed track they can use only the data provided by the organizers. This data includes monolingual corpora for each language which should be used for the mining of translations. Furthermore, the shared task provides training data that consists of tab-separated bilingual word pairs divided into high, medium and low frequency groups, i.e., words ranking in 5000 most frequent words, in the range of $5001 - 20000$ and $20001 - 50000$ respectively. The test sets are also split in the three groups, with 2000 words each. Both train and test are a subset of the MUSE dictionaries (Conneau et al., 2018) which were created using a Facebook internal translation tool. In addition they take the polysemy of words into account, meaning that some words have multiple translations. Due to this, the performance of the systems is determined by computing precision, recall and F_1 score[1] instead of *acc@n* used in other works (Vulic and Korhonen, 2016). For further information about the official data and setup we refer to the shared task description paper (Rapp et al., 2020).

3. Approach

To solve the BDI task we rely on both BWE and word surface based similarity. As in many related works, we calculate the vector similarity of words in order to find target language words having similar meaning compared to a given input word. However, BWEs tend to perform poorly when translating named entities and low-frequency words (Braune et al., 2018; Riley and Gildea, 2018). To alleviate the problem, we follow the approach of (Braune et al., 2018) and combine word similarity information from multiple BWE models and we look for similarly written source and target language words. The latter can be solved by looking for orthographically similar words in the case of English and German. On the other hand, for English and Russian the approach is not applicable due to the different character sets of the two languages, thus we employ an unsupervised transliteration model.

3.1. Bilingual Word Embeddings

To build BWEs we follow the mapping approach of (Mikolov et al., 2013b), i.e., we build monolingual word embeddings (MWEs) which we then align to a share space using a seed dictionary. We create 4 types of MWE models for each language, since it was shown that combining them is beneficial for BDI (Braune et al., 2018): $\{word2vec, fasttext\} \times \{cbow, skipgram\}$ (Mikolov et al., 2013a; Bojanowski et al., 2017). We perform the mapping using *VecMap* (Artetxe et al., 2018) which learns an orthogonal projection of the source MWE to the target space. Although the approach supports unsupervised mapping, we use it in a supervised setup. As the seed lexicon, we use part of the provided high frequency dictionary. Although the dictionary contains multiple translations for some source words, we only use the first translation of each word in order to reduce noise. Finally, we generate a *similarity dictionary* based on each BWE type containing translation candidates, i.e., the 100 most similar target language words, for each source language word along with their similarity scores. We calculate the cosine similarity based *Cross-Domain Similarity Local Scaling* (CSLS) metric as the similarity score (Conneau et al., 2018) which adjusts the similarity values of a word based on the density of the area where it lies, i.e., it increases similarity values for a word lying in a sparse area and decreases values for a word in a dense area. In the simple case, word translation could be done by using the most similar target candidate for a given source word based on one of the dictionaries. On the other hand, our aim is to exploit the advantages of all BWE types which we achieve by ensembling the generated similarity dictionaries.

Ensembling In order to merge various similarity dictionaries we follow a similar approach as (Braune et al., 2018). For this, we create a final similarity dictionary containing the 100 most similar target words for each source word along with their ensembled similarity scores which is given by:

$$Sim_e(S,T) = \mathcal{Q}_{i=1}^{M} \gamma_i Sim_i(S,T) \qquad (1)$$

where S and T are the source and target words, $Sim_i(\cdot,\cdot)$ and γ_i is the similarity of two words based on the i^{th} BWE type and its weight. As the $\mathcal{Q}$ function, we experimented with summing the weighted values or taking their maximum value. The former aims to emphasise candidates that are ranked high by multiple models while the latter takes the candidates in which a given model is confident. For simplicity we only calculate the score for target words that are in any of the dictionaries for a given source word instead of the full target language vocabulary. If a candidate word T is not in dictionary i we set $Sim_i(S,T)$ to 0. γ_i are tuned on the development set.

The above equation only requires dictionaries containing word pairs and their similarities allowing us to employ information from other sources as well, such as orthography and transliteration which we discuss in the following.

[1] F_1 is the official score for system ranking.

3.2. Orthographic Similarity

The translation of many words, such as named entities, numerical values, nationalities and loan words, are written similarly as the source word, thus we rely on orthographic similarity to improve the translation of such words. For English and German we follow the approach of (Braune et al., 2018) and use Levenshtein similarity, more precisely one minus the normalized Levenshtein distance, as the orthographic similarity of a given word pair. We generate similarity dictionaries as before but containing orthographically similar words, which we use as an additional element during ensembling. The generation of such a dictionary is computationally heavy, since each source word has to be compared to each word in the target language vocabulary leading to a large number of word pairs. Since most of the word pairs are not orthographically similar we follow the approach of Riley and Gildea (2018) to reduce the number of word pairs to compare. For this the *Symmetric Delete* algorithm is used, which takes as arguments a list of source words, target vocabulary and a constant k, and identifies all source-target word pairs that are identical after k insertions or deletions. We then calculate the Levenshtein similarity only for such word pairs.

3.3. Transliteration score

When dealing with word pairs from different scripts (i.e. En-Ru), we need a different measure of similarity because the alphabets are not shared. If we consider rare words, we know that many of them are transliterated (e.g., translated preserving the sound). Adam/Адам and Laura/Лаура are example of English-Russian transliteration pairs. Therefore, we propose a new method to capture similarities between words from different scripts through transliteration scores. In particular, we aim to improve the BWEs for rare and less frequent words incorporating the word scores coming from our transliteration model. The method is unsupervised given that we do not have transliteration pairs for training in the shared task setup – we have translation pairs, but they are not annotated as transliteration vs non-transliteration. The model is used in an unsupervised way to clean the training set and to get the final predictions. Our method consists of training a sequence-to-sequence model (Sutskever et al., 2014) on a "cleaned" set to get the transliteration scores. The model and the cleaning process are explained in the following.

3.3.1. Transliteration model

Once we cleaned the whole dataset as explained in the section below, we use it as the training set for our seq2seq model. The model works at the character-level and is made of an encoder and a decoder part with attention. They both contain multi-layered Gated Recurrent Units (Cho et al., 2014) but the encoder uses bidirectional GRUs that is able to encode both past and future context. The decoder exploits the "Global Attention" mechanism with the "dot" method of (Luong et al., 2015) to diminish the information loss of long sequences. The model has one encoder and one decoder layer with hidden size of 128. We use a dropout regularization probability of 0.1 and a learning rate of 0.01 with the SGD optimization algorithm.

Once the model is trained, we use it to calculate the negative log likelihood probability (pNLL) of each word in the target language vocabulary with respect to each test word because we saw that it was working better than the generation of transliteration words. In this way, we generated the similarity dictionary and we selected the 100 top scored words. Given a word pair $[S, T]$ with $t_1, .., t_N \in T$, we define the score as:

$$pNLL = \frac{(\sum_{i=1}^{N} nll(t_i)) + nll(EOS)}{N+1} \qquad (2)$$

where $nll(t_i)$ is the Negative Log Likelihood probability of the i^{th} character in T, and EOS is the "End Of String" token.

3.3.2. Cleaning process

The cleaning process aims to reduce the number of non-transliteration pairs in the initial dataset in an unsupervised way to better train the final transliteration model. The dataset is considered "cleaner" if it contains less non-transliteration pairs than the initial one and still enough transliteration pairs to allow the training of the model.

First, we randomly select 10 pairs that have a length difference greater than one as the "comparison set" and we fixed it for all the cleaning process. This length difference helps to find pairs that in most cases are not transliteration.

We then carry out an iterative process. We split the dataset in training and test sets (80%-20%) and we train the character-level Encoder-Decoder model, explained in section 3.3.1 above, on the training set. The number of steps was chosen based on previous experiments. Then, we evaluate the test set on the model and we obtain a score for each test pair $(source, target)$. A score measures the negative log likelihood probability of predicting the target given the input. Higher scores mean higher probability for the input and target to be transliterations of each other. Then, we calculate the scores for the comparison set in the same way and we remove all the test pairs that are below the average score of the comparison set. Finally, we shuffle the training set with the remaining test pairs and we divide again in training and test. We repeat this process training a new model every time and cleaning the test set for a fixed number of iterations found experimentally,

The dataset has been divided into low, medium and high-frequency pairs. We exploited this fact with the assumption that the low-frequency set should contain rare words and more nouns, so consequently more transliteration pairs than the high-frequency set. Therefore, we first clean the low set with the iterative process. Then, we mix the cleaned low set with the uncleaned medium set and run the process on it. Finally, we mix the result of this process with the high-frequency set and run the last iterative method to get the cleaned dataset that we used in the final transliteration model. Note that we only rely on the training portion of the released high, medium and low dictionaries (see Section 4).

3.4. Dynamic Translation

BDI is often performed by returning the top-1 or top-5 most probable translations of a source word (Mikolov et al., 2013b). Since the dictionaries of the shared task contain a dynamic number of translations, the participants had to decide the number of words to return. During our experiments we found that using top-1 translation for the low and middle and top-2 for high frequency sets gives consistent results thus we used this solution as our official submission. However, we experimented with dynamic methods as well. Based on the manual investigation of the ensembled word pair similarity scores, we found that having a global threshold value would not be sufficient for selecting multiple translations for a given source word, since the similarity values of the top-1 translations have a large deviation across source words. This is also known as the hubness problem (Dinu and Baroni, 2014), i.e., the vector representation of some words tend to lie in high density regions, thus have high similarity to a large number of words, while others lie in low density regions having low scores. Instead of using a global threshold value, we followed the margin based approach proposed by (Artetxe and Schwenk, 2019) for parallel sentence mining which in a sense calculates a local threshold value for each source word. We adapt this method for BDI and calculate a score of each candidate word T for a given source word S by:

$$score(S, T) = margin(Sim_e(S, T), avg(S)) \quad (3)$$

where $avg(S)$ is the average similarity scores of S and the 100 most similar candidates based on the ensemble scores $Sim_e(\cdot, \cdot)$. We experimented with two variants of the $margin$ function:

$$marginDistance(x, y) = x - y \quad (4)$$

$$marginRatio(x, y) = \frac{x}{y} \quad (5)$$

The aim of both methods is to normalize the similarities based on the averaged similarity values so that a global threshold value can be used to select translations. The former method calculates the distance between the similarity value of the target candidate and the averaged similarity while the latter calculates their ratio. Finally, we consider each target candidate of a given source word as translation if its score is higher than the threshold value. We tune one threshold value for each language pair and word frequency category using the development sets. In addition, since each source word should have at least one translation, we always consider the top-1 most similar candidate to be a translation.

4. Experimental Setup

We submitted BDI outputs for both the closed and open tracks which differ only in the used BWEs. For the closed track we only relied on the released monolingual corpora and training dictionaries. For the MWEs we used the *WaCKy* corpora (Baroni et al., 2009) and built *word2vec* cbow and skipgram models (Mikolov et al., 2013a), and *fasttext* cbow models (Bojanowski et al., 2017), while we used the released fasttext skipgram models from the shared task website. We used the same parameters used by the organizers for both methods: minimum word count 30; vector dimension 300; context window size 7; number of negatives sampled 10 and in addition, number of epochs 10 for fasttext. To align MWEs of the same type, we used *VecMap* (Artetxe et al., 2018) in a supervised setup. As the training signal we used the official shared task dictionaries which are a subset of the *MUSE* dictionaries released in (Conneau et al., 2018). We split them into train, development and test sets (70%/15%/15%)[2] which we used for training BWEs and the transliteration model, tuning parameters and reporting final results respectively. Since we tuned various parameters, such as ensembling weights or threshold values for margin based translation, for each language pair and frequency category, we do not report each value here but discuss them in the following section. For the generation of BWE based similarity dictionaries we only considered the most frequent $200K$ words when calculating CSLS similarities as in (Conneau et al., 2018). We experimented with larger vocabulary sizes but achieved lower scores. In contrast, for the orthography and transliteration based dictionaries we considered all words in the monolingual corpora which have at least frequency 5[3].

For the open track we followed the same approach as above but instead of using WaCKy based MWEs we used pre-trained Wikipedia based monolingual fasttext skipgram models similarly as in (Conneau et al., 2018). Although we use only one type of BWE model (instead of four) in addition to the orthography or transliteration based similarities we achieved higher performance especially for the middle and low frequency sets.

5. Results

As the official evaluation metric of the shared task we present F_1 scores of our approach. We compare multiple systems to show the effects of various modules of our approach on our test splits in Table 1. We compare systems using only one similarity dictionary using either fasttext (FTT) cbow or surface similarity and our complete system ensembling five similarity dictionaries using tuned weights (two for the open track). We also show results of our open track submission (Wiki). All systems return top-n translations except *ensemble + margin*. We used $n = 1$ for the low and middle frequency sets and also for Ru-En high, while for the rest $n = 2$ gave the best results. When using margin based translation, we show the best performing method based on the development set which we discuss in more details below. In general, it can be seen that in our closed track submission the best results were achieved by ensembling various information from different sources. The BWE based model achieved fairly good results for the high and middle frequency sets but often lower results than the surface similarity based model for low frequency words. On the contrary, the surface based systems performed well as the frequency of words decreases, having low scores for the high set. Based on investigation of the test splits, not surprisingly the results correlate with the number of words that are written similarly on both the source and target language sides showing the importance of this module during BDI.

[2] We kept all translations of a given source word in the same set.

[3] Additionally, we filtered words that contained at least 2 consecutive punctuation marks or numbers.

	High			
	En-De	De-En	En-Ru	Ru-En
FTT cbow	38.17	46.37	33.52	46.78
Surface	4.31	3.41	7.38	14.64
Ensemble	40.59	49.56	38.33	54.12
Ensemble + Margin	39.76	49.90	36.23	54.71
Wiki	41.40	48.61	39.43	54.90

	Middle			
	En-De	De-En	En-Ru	Ru-En
FTT cbow	30.62	36.00	20.14	39.82
Surface	7.76	10.11	13.47	16.93
Ensemble	47.76	51.71	33.24	49.64
Ensemble + Margin	47.76	51.89	36.17	49.72
Wiki	49.18	53.66	43.55	56.53

	Low			
	En-De	De-En	En-Ru	Ru-En
FTT cbow	24.19	33.05	15.03	21.53
Surface	24.62	20.12	20.62	30.25
Ensemble	63.82	69.41	30.11	42.99
Ensemble + Margin	63.82	69.41	30.50	43.17
Wiki	65.14	73.10	51.72	57.01

Table 1: F_1 scores for English-German and English-Russian language pairs in both directions and the three frequency categories on our test split. The first two models use either a dictionary based on embeddings or surface similarity while the rest combines all of the available (two for Wiki and five for the rest). Ensemble + Margin shows results with dynamic number of translations per source words using the best margin based method and top-n ($n \in \{1, 2\}$) is applied for the rest. Wiki shows our open track submission.

By looking at the ensembling scores, the BWE and surface scores seem additive showing that the two methods extend each other, i.e., the source word could be translated with either of the models.

Model weights As mentioned, we tuned our system parameters on the development set. Without presenting the large number of parameters, we detail our conclusions. Comparing the usefulness of the BWE types we found similarly to (Braune et al., 2018) that fasttext models are more important by handling morphological variation of words better due to relying on character n-grams which is especially important for Russian. On the other hand, word2vec models also got significant weights showing their additional positive effect on the results. Comparing skipgram and cbow models we found that the weights of fasttext cbow and fasttext skipgram are similar (the former has a bit higher weight) while word2vec cbow got close to zero weight, only the word2vec skipgram model is effective. The weights of the surface based similarity dictionaries were lowest for the high frequency sets and higher for the other two, but counter intuitively it was the highest for the middle set 3 out of 4 times. The reason for this is that many words in the low sets are not included in the most frequent $200K$ words that we used in the BWEs but in the surface dictionaries only, thus independent of the weights the translation is based on the latter. On the other hand, many source words have similarly written pairs on the target side even though they have proper translations, e.g., source: *ambulance*; transliteration: *амбуланс*; translation: *скорая*, thus having high weight led to incorrect translations. As mentioned in Section 3 we experimented with summing the scores in the dictionaries during ensembling or taking their maximum. The former consistently performed better for En-De and De-En while the latter performed better for En-Ru and Ru-En. The reason lies in the different surface models: orthographic similarity for German and transliteration for Russian.

Dynamic translation The ensemble+margin system shows our results with the system predicting a dynamic number of words as translation based on the margin method. We tuned the threshold value for both *marginDistance* and *marginRatio* and show the best performing setup. We achieved some improvements in most of the cases compared to ensemble with top-n, except for En-De high and En-Ru high. On the other hand, we achieved significant improvements for En-Ru middle and Ru-En low. However, we found that this method is not robust in various scenarios since the best parameters (margin method variation and threshold value) were different across our test sets and we found no pattern in them, e.g., high threshold for low frequency sets and low value for higher frequencies. On the other hand, top-1 and top-2 translations performed more consistently. We expect the margin based method to perform better than top-n for mixed frequency test set.

Open Track In our open track submission we ensembled Wikipedia based fasttext skipgram based BWEs with surface information. Although our system relied only on the two similarity models we achieved significant improvements compared to our closed track systems, especially for En-Ru and Ru-En. The reason for this lies in the number of OOVs in the BWE vocabularies. As mentioned we used the $200K$ most frequent word for both WaCKy and Wikipedia based BWEs but for the former more source test words are OOVs. We investigated the gold translations as well and found a similar trend, i.e., there are more cases for the closed track models where the source word's embedding is known but not that of its gold translation. Our conjecture is that the machine translation system used for the creation of the MUSE dictionaries relies more on Wikipedia texts, thus these models perform better on these test sets.

Manual analysis In table 2 we show interesting samples taken from test set results that we created out of the training data provided. The last two columns show the top predictions according to BWE based scores, and orthographic or transliteration scores. The Surface column is chosen as the final prediction when no translation is provided for the source word meaning that the source is not present in the BWEs. This helps to solve OOV word issues. We can see that the surface prediction is also useful for source words that are not proper names like in the *[polarität, polarity]* example. The last two rows show negative results where the ensembling led to incorrect predictions. The *[бартольд, barthold]* sample shows an incorrect weighting of the final prediction which for example could have been solved with a local weighting that could adjust the importance of the

Source	Gold	Ensemble	FTT cbow	Surface
фейерверки	fireworks	fireworks	**fireworks**	feierwerk
левандовский	levandovski	levandovski	/	**levandovski**
workouts	тренировки	тренировки	**тренировки**	воркуты
hippocrates	гиппократ	гиппократ	**гиппократ**	покравительство
massimiliano	массимилиано	массимилиано	/	**массимилиано**
bolschoi	bolshoi	bolshoi	/	**bolshoi**
nikotin	nicotine	nicotine	alcohol	**nicotine**
polarität	polarity	polarity	polarities	**polarity**
бартольд	barthold	ismaili	ismaili	**barthold**
inedible	ungenießbar	incredible	**ungenießbar**	incredible

Table 2: Samples from our test set. The *Ensemble* column contains the output of our complete system, *FTT cbow* contains the output based on FTT only, and *Surface* column contains the output based on the orthographic or transliteration similarity scores. In bold there are the correct predictions in the last two columns. The slash *"/"* symbol indicates that the source word is not in the embedding vocabulary. The last two samples are cases where the ensemble model selected the final prediction wrongly.

	High			
	En-De	De-En	En-Ru	Ru-En
Closed	41.7	46.8	39.4	54.2
Open	42.0	46.6	38.2	56.2
	Middle			
	En-De	De-En	En-Ru	Ru-En
Closed	45.6	53.8	34.4	51.5
Open	47.9	57.9	40.4	56.9
	Low			
	En-De	De-En	En-Ru	Ru-En
Closed	66.0	69.2	29.9	41.4
Open	67.1	72.9	49.2	58.4

Table 3: Official BUCC 2020 results of our closed and open track submissions.

BWEs and transliteration based on the candidate scores. The last sample is incorrect probably because of the strong similarity between the source word and the orthography top prediction. We also have noise issues in this case (i.e., "incredible" is not a German word) that could be solved with a language detection based filtering.

Official results We show the performance of our submissions in the official shared task evaluation in table 3. Overall, our system was ranked in the top 3 teams and it achieved top 1 results on the English and Russian language pairs. As mentioned above our closed track submission involved the ensembling of BWE and word surface similarity scores and taking either top-1 or top-2 translations based on the frequency set. The open track submission differs only in the used word embeddings, e.i., we used pre-trained wikipedia fasttext skipgram embeddings only. Our official results are similar to the results on our test splits in table 1 which indicates the robustness of our approach.

6. Conclusion

Bilingual dictionary induction is an important task for many cross-lingual applications. In this paper we presented our approach to the BUCC 2020 which is the first shared task on BDI aiming to compare various systems in a unified framework on multiple language pairs. We followed a BWE based approach focusing of low frequency words by improving their translations using surface similarity measures.

For our English-German system we used orthographic similarity. Since for the English-Russian language pair orthography is not applicable due to different scripts, we introduced a novel character RNN based transliteration model. We trained this system on the shared task training dictionary which we cleaned by filtering non-transliteration pairs. In our results we showed improvements compared to a simple BWE based baseline for high, medium and low frequency test sets. We showed that by using multiple BWE types better performance can be reached on the high set. Furthermore, the medium and low sets surface similarity gave significant performance improvements. In addition to translating words to their top-1 or top-2 most similar candidates, we experimented with a margin based dynamic method which showed further improvements. On the other hand, since we found that it is not robust across the various setups, we used top-n translations in our official submission. Based on the analysis of our results, future improvement directions are better combinations of various similarity dictionaries, such as source word based local weighting, getting rid of the seed dictionary in the overall method, and a more robust dynamic prediction approach.

Acknowledgements

We gratefully acknowledge funding for this work by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreements № 640550 and № 740516).

Bibliographical References

Artetxe, M. and Schwenk, H. (2019). Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203.

Artetxe, M., Labaka, G., and Agirre, E. (2018). 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.

Artetxe, M., Labaka, G., and Agirre, E. (2019). An Effective Approach to Unsupervised Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Braune, F., Hangya, V., Eder, T., and Fraser, A. (2018). Evaluating bilingual word embeddings on the long tail. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 188–193.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). 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)*, pages 1724–1734.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2018). Word Translation Without Parallel Data. In *Proceedings of the International Conference on Learning Representations*, pages 1–14.

Czarnowska, P., Ruder, S., Grave, E., Cotterell, R., and Copestake, A. (2019). Don't Forget the Long Tail! A Comprehensive Analysis of Morphological Generalization in Bilingual Lexicon Induction. 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)*, pages 974–983.

Dinu, G. and Baroni, M. (2014). Improving zero-shot learning by mitigating the hubness problem. *CoRR*, abs/1412.6568.

Huck, M., Hangya, V., and Fraser, A. (2019). Better OOV Translation with Bilingual Terminology Mining. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5809–5815.

Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-Based & Neural Unsupervised Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting Similarities among Languages for Machine Translation. *CoRR*, abs/1309.4.

Rapp, R., Zweigenbaum, P., and Sharoff, S. (2020). Overview of the fourth BUCC shared task: bilingual dictionary extraction from comparable corpora. In *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 1–6.

Riley, P. and Gildea, D. (2018). Orthographic Features for Bilingual Lexicon Induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 390–394.

Riley, P. and Gildea, D. (2020). Unsupervised bilingual lexicon induction across writing systems. *arXiv preprint arXiv:2002.00037*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Vulic, I. and Korhonen, A. (2016). On the Role of Seed Lexicons in Learning Bilingual Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 247–257.

Chapter 3

Corresponds to the following publication:

> **Silvia Severini**, Viktor Hangya, Alexander Fraser, and Hinrich Schütze. "Combining Word Embeddings with Bilingual Orthography Embeddings for Bilingual Dictionary Induction.". In Proceedings of the 28th International Conference on Computational Linguistics. 2020.

Declaration of Co-Authorship: I conceived the original research contribution. I designed all the models and performed all the evaluations except for the VecMap-related evaluations (Viktor Hangya). I regularly discussed this work with my advisor Hinrich Schütze. I wrote the initial draft of the article and did most of the subsequent corrections. All authors helped review the final draft of the paper and gave advice.

Combining Word Embeddings with Bilingual Orthography Embeddings for Bilingual Dictionary Induction

Silvia Severini, Viktor Hangya, Alexander Fraser, Hinrich Schütze
Center for Information and Language Processing
LMU Munich, Germany
{silvia, hangyav, fraser}@cis.uni-muenchen.de

Abstract

Bilingual dictionary induction (BDI) is the task of accurately translating words to the target language. It is of great importance in many low-resource scenarios where cross-lingual training data is not available. To perform BDI, bilingual word embeddings (BWEs) are often used due to their low bilingual training signal requirements. They achieve high performance, but problematic cases still remain, such as the translation of rare words or named entities, which often need to be transliterated. In this paper, we enrich BWE-based BDI with transliteration information by using *Bilingual Orthography Embeddings* (BOEs). BOEs represent source and target language transliteration word pairs with similar vectors. A key problem in our BDI setup is to decide which information source – BWEs (or semantics) vs. BOEs (or orthography) – is more reliable for a particular word pair. We propose a novel classification-based BDI system that uses BWEs, BOEs and a number of other features to make this decision. We test our system on English-Russian BDI and show improved performance. In addition, we show the effectiveness of our BOEs by successfully using them for transliteration mining based on cosine similarity.

1 Introduction

The task of Bilingual Dictionary Induction is defined as finding target language translations of source language words. It is an important building block in the area of Machine Translation (MT) and it is one of the main tasks for bilingual word embedding evaluation (Mikolov et al., 2013b; Vulic and Korhonen, 2016). Recent work shows that good performance can be achieved relying only on BWEs, which can be built with only a weak bilingual signal, such as a small seed lexicon of a few thousand word pairs (Mikolov et al., 2013b) or common tokens in the source and target languages (Artetxe et al., 2017). In addition, they can even be built without any bilingual signal (Conneau et al., 2018; Artetxe et al., 2018), making them the basis of unsupervised MT systems (Lample et al., 2018; Artetxe et al., 2019).

Standard BDI learns word representations based on approaches that exploit solely word-level information such as *word2vec* (Mikolov et al., 2013a) or *fasttext* (Bojanowski et al., 2017) and then map them to a shared BWE space. Although BWE-based approaches show high BDI performance, they struggle with a subset of hard-to-translate words such as named entities for which orthographic information should be used instead of semantic information. Several approaches have integrated orthographic information into the BDI system. Heyman et al. (2017) relied on character-level information in their classification based BDI system by using an RNN architecture. Braune et al. (2018) combined orthographic information with BWE-based word similarity information using an ensembling method. Both of these approaches showed improved results, but they relied on Levenshtein distance to get translation candidates for a source word during prediction, which is not applicable for language pairs with different scripts. To bridge the gap between languages with different scripts, a transliteration system was employed by Severini et al. (2020). They followed the approach of Braune et al. (2018) but used the transliteration system instead of Levenshtein distance to get candidates used in the ensembling model. On the other hand, as they also showed,

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 28th International Conference on Computational Linguistics, pages 6044–6055
Barcelona, Spain (Online), December 8-13, 2020

the ensembling approach often fails to decide correctly if a word has to be transliterated or not; this is because there are only two independent scores available, the score of the (semantic) BWE, and the score of the (orthographic) transliteration model.

In this paper, we present our novel approach to BDI focusing on words that have to be transliterated to another script, which is especially important for low-frequency words, but also relevant for high-frequency named entities. Our aim is to improve BDI systems in two respects: (i) eliminating the need for language specific orthographic information, such as is used in Levenshtein distance, and (ii) to be able to better decide when to choose transliteration over semantic translation. We propose a new approach for language pairs with different scripts by combining semantic information with orthographic information. For the latter we introduce *Bilingual Orthographic Embeddings* (BOEs) of words, which represent transliteration pairs in the source and target language with similar vectors. We build BOEs using a novel transliteration system trained jointly for both language directions. We refer to this novel system as *seq2seqTr*. *seq2seqTr* can also be used to extract candidate transliterations for a source word. It is applicable to any language pair (as opposed to Levenshtein distance). To make a more informed decision about which words should be transliterated (which means we shoud primarily trust the BOEs) and which should be semantically translated (which means we should primarily trust the BWEs), we use a classification approach similar to (Heyman et al., 2017), exploiting our pretrained encoder from seq2seqTr. In contrast to their approach, we use additional features, such as frequency, length, similarity scores, and the ranks assigned by the semantic and character-level submodels, and show that they are necessary to make the right decision.

We test our system on the *English-Russian* (En-Ru) data provided in the BUCC 2020 shared task (Rapp et al., 2020). Test dictionaries were released in three frequency categories: high, middle and low. We evaluate our system on all three sets, both separately and jointly, and show improved performance on all three frequency ranges compared with previous approaches. Furthermore, we show that our classification system is more robust than the ensembling of Severini et al. (2020), which required specialized tuning on each frequency set. Lastly, we conduct a further analysis of the quality of the proposed BOEs by running transliteration mining on the NEWS 2010 shared task data (Kumaran et al., 2010) by using the vector similarity of Bilingual Orthographic Embeddings of words. We show good performance on the task indicating the usefulness of BOEs for other downstream tasks.

2 Related Work

2.1 Bilingual Dictionary Induction

BWEs are often used for solving BDI tasks by calculating cosine similarity of word pairs and taking the n most similar target candidates as translations for a source word. As opposed to general MT based approaches that rely on parallel sentences, BWEs are also effective when only a small seed lexicon is provided (e.g., (Mikolov et al., 2013b)). Conneau et al. (2018) and Artetxe et al. (2018) dispense with seed dictionaries and iteratively improve the mapping from an initial weak solution in a self-learning approach. This setting provides a building block for unsupervised MT and is particularly effective in the low-resource setting where less parallel seed data is available (Artetxe et al., 2019; Lample et al., 2018).

BWE-based methods perform worse for low frequency words due to poor vector representations (Braune et al., 2018; Riley and Gildea, 2018; Czarnowska et al., 2019). Koehn and Knight (2002) and Haghighi et al. (2008) show that orthographic features help in the translation process. Languages with a common alphabet (e.g., English/German) often have word pairs with similar orthography (e.g., *Concepts/Konzepte*, *Philosophies/Philosophien*), especially in the case of low frequency words. Riley and Gildea (2018) integrate orthographic information into the vector representation of such words and into the mapping procedure of BWEs to improve their quality. Braune et al. (2018) use character n-gram representations and Levenshtein distance to improve BDI while Heyman et al. (2017) extract this feature automatically from training data. In languages with different scripts (e.g., English/Russian), the source word is often written with the closest corresponding letters of the target alphabet, i.e., it is transliterated. *Richard/*Ричард and *integrator/*интегратор are examples of transliterations between English and Russian. Irvine and Callison-Burch (2017) applied Levenshtein distance to language pairs with different alphabets by first transliterating from non-Latin to Latin scripts. In contrast, we use a novel transliteration model that encodes the relevant information directly into BOEs without requiring a separate transliteration step.

2.2 Transliteration

As motivated above, transliteration mining is an important task that bridges the gap between languages with different scripts. The NEWS transliteration shared task has been a continuous effort to promote research on this task since 2009 (Li et al., 2009; Chen et al., 2018). A fully unsupervised transliteration model was proposed by Sajjad et al. (2017); it consists of interpolated statistical sub-models for transliteration and non-transliteration detection. In this work we follow a similar idea and propose an unsupervised neural network based system to look for transliteration pairs of source words as possible translation candidates. We also use this system to build BOEs of words.

In a parallel sentence mining approach, Artetxe and Schwenk (2019) use a shared encoder and decoder for all languages to build a language agnostic sentence encoder. They use the encoder representations as sentence embeddings to efficiently mine parallel sentences. Similarly, our BOEs are extracted from a single language agnostic encoder, for both English and Russian. In an ablation study, we check the quality of our BOEs on the NEWS 2010 shared task (Kumaran et al., 2010); see below.

3 Approach

To tackle the BDI task we exploit BWEs, character-level information (in the form of BOEs) and manually engineered features – such as word frequency and length – and integrate them into a classifier that predicts if the word pair is a translation or not. We extract candidate translations for a source word based on (i) BWEs and (ii) *seq2seqTr* (our transliteration model) as described in the following sections. Finally, we rerank the two groups of candidates with our classification system described below and take the top ranked candidates as our prediction.

3.1 Bilingual Word Embeddings

To create BWEs we use monolingual word embeddings (MWEs), learned with *fasttext skipgram* (Bojanowski et al., 2017), and align them to a shared space with a seed dictionary that consists of high frequency word pairs using the approach of Artetxe et al. (2018). We then generate translation candidates for each source word by taking the n target language words that are the most similar. Our similarity measure is the cosine-similarity-based *Cross-Domain Similarity Local Scaling* (CSLS) measure (Conneau et al., 2018). We use these target candidates along with the corresponding source words as classification samples during prediction.

3.2 The Transliteration Model: seq2seqTr

In this work, we focus on two languages that have different alphabets since our aim is to improve application scenarios in which the Levenshtein distance is not applicable. To address language pairs with different scripts and pairs of infrequent words such as named entities, we propose seq2seqTr, a novel transliteration system. seq2seqTr is trained on a list of word pairs that are translations of one another – both transliterations and non-transliterations. It is unsupervised because we do not rely on labels to distinguish between transliteration and non-transliteration training pairs. seq2seqTr is a character-level sequence-to-sequence model (Sutskever et al., 2014) with a single-layer encoder and a single-layer decoder. The encoder is a bidirectional GRU (Cho et al., 2014) while the decoder is unidirectional with attention (Luong et al., 2015). The input characters are represented as vectors. Figure 1 (bottom) depicts the model. We use this model to calculate the probability of each word in the target language vocabulary with respect to each source test word. The probability corresponds to the average negative log likelihood of the characters in the target word with respect to the source word. We select n target transliteration candidates for each source word.

To train seq2seqTr, we start with the same training dictionary as for building BWEs. Since it contains many non-transliteration pairs, we reduce their number with an iterative cleaning process. The dictionary is considered “cleaner” if it contains fewer non-transliteration pairs than the initial one; but it should still have enough transliteration pairs to allow the effective training of the transliteration model. In more detail, we first select 100 pairs randomly from the training dictionary. We call this the *comparison set*. We limit the sampling to pairs that have the same length to make sure that a sufficient number of transliterations is in the comparison set. Then, we split the dictionary[1] into train and test (80%-20%) and we train seq2seqTr on this new training set with early stopping. We calculate the 95% confidence interval for the scores (average log likelihood, see above) of the comparison set. Let θ be the upper bound of the confidence interval. We then remove the pairs in the secondary test set that have a score lower than θ. Finally, we merge secondary training set and cleaned secondary test set, shuffle them and iterate the process until we can no longer remove pairs. The intuition for the choice of θ is that we only want to keep pairs in the final training set that are transliterations with high probability – pairs whose scores are clearly higher than the typical scores in the comparison set (i.e., are larger than θ) should have this property. The same iterative method is run to clean the development set portion of the dictionary.

3.3 Bilingual Orthography Embeddings

As a representation that is informative about orthography of source and target words, we build Bilingual Orthography Embeddings (BOEs). The BOE space is a common representation space – just like the BWE space – and transliteration pairs are represented with similar vectors. We again use the same training set (the cleaned training set) and our seq2seqTr architecture, but we tune GRU hidden and character embedding sizes for the BDI task. More importantly, we employ a slightly different training procedure to tie the two languages together and to build a language agnostic encoder that works for both source and target languages. To this end, we train seq2seqTr with source-to-target and target-to-source word pairs where we indicate the output language using a language specific marker at the first position of the target decoder output. Since we want a language agnostic encoder, we do not use such a marker on the encoder side.

In addition to source-to-target and target-to-source word pairs, we also train seq2seqTr on source-to-source and target-to-target pairs, i.e., we also train the model as an autoencoder. We use the same words for within-language as for cross-lingual pairs. Without training seq2seqTr to be an autoencoder, the encoder representations for Russian and English would not be in the same subspace. As the BOE representation of a word, we take the final hidden state of the encoder GRU layer since it is also used to initialize the decoder of the complete transliteration model. We also experimented with taking the averaged output states of the final encoder layer as BOEs and decoder initialization, similar to (Artetxe and Schwenk, 2019), but it gave slightly worse results.

3.4 BDI Systems

We first introduce our baselines and then explain our classification model.

Ensembling Baseline Severini et al. (2020) used ensembling to combine the candidate translations for a source word coming from BWEs and a transliteration module. The combination score of source word s and translation candidate t is a simple weighted sum:

$$\mathrm{sim}(s,t) = \gamma_{\mathrm{bwe}}\mathrm{sim}_{\mathrm{bwe}}(s,t) + \gamma_{\mathrm{tr}}\mathrm{sim}_{\mathrm{tr}}(s,t) \tag{1}$$

Here, $\mathrm{sim}_{\mathrm{bwe}}(s,t)$ is the CSLS similarity of the word pair, $\mathrm{sim}_{\mathrm{tr}}(s,t)$ is a similarity score based on the probability of t being the transliteration of s based on the transliteration model (see (Severini et al., 2020) for details), and the γ_i are weights of the two modules tuned on dev.

The downside of this approach is that it is strongly dependent on the order of words in the candidate lists of BWEs and transliterations. If the correct translation is contained in one of the lists but not at the first position it will not be picked as the output translation. Equation 1 has an implicit reranking capability, i.e., if a word is contained in both BWE and transliteration lists their scores get summed, which can move it higher in the final ensembled list compared to words that are present in only one

[1] Note that we use only the training portion of our dictionary for the three frequency sets (see Section 4) for this process, i.e., we split the complete training dictionary into a secondary training set and a secondary test set.

list. On the other hand, this reranking is limited as we show in our experiments. Moreover, a simple linear combination, only based on the final scores of the two modules, does not give enough flexibility to integrate other features that could help decide if a lower ranked candidate should be picked as the final translation. We aim to overcome these problems in our proposed system.

Classification Baseline The second baseline is Heyman et al. (2017)'s approach. It is a neural classifier that takes word-level information and character-level information as input. The system has two parts. The first is a character-level RNN that aims to learn orthographic similarity of word pairs. At time-step i of the input sequence, it feeds the concatenation of the i^{th} character embedding of the source and target language words to the RNN layer. The second part is the concatenation of the BWEs of the two words learned independently of the model; based on this the model aims to learn the similarity of the two semantic representations. Dense layers are applied on top of the two modules before the output softmax layer. The classifier is trained using positive and negative word pair examples. Negative examples are randomly generated for each positive one in the training lexicon. However, since characters at the same time steps are compared in the RNN module, transliteration word pairs with 1-to-many character correspondences are hard to handle correctly – any shift in the alignment then affects subsequent pairs of letters. Furthermore, apart from the source and target BWEs and their interpolated character string representation (given by the RNN module), the feed-forward layer has no information to decide whether a source word should be translated based on transliteration or based on BWEs.[2]

Classification Model Similar to Heyman et al. (2017), we employ a classification approach. Our classifier takes as input BWEs, orthography (in the form of BOEs) and additional features.

Figure 1 shows the model. It has two fully-connected feed forward layers with dropout and non-linear activation function. Its input consists of the following: BWEs of the source and target words ($\text{bwe}_s, \text{bwe}_t$), their BOEs ($\text{boe}_s, \text{boe}_t$), log frequency values (f_s, f_t) and their absolute difference, log length values (l_s, l_t), the similarity value of source and target BWEs (sim_{bwe}), the conditional probability $P(t|s)$ computed by seq2seqTr (which we call sim_{tr}) and the log of the position of the candidate in the candidate list (e.g., 1st, 2nd, etc.) for BWEs and for seq2seqTr ($\text{pos}_{\text{bwe}}, \text{pos}_{\text{tr}}$). The intuition behind f_s and f_t is that transliteration happens more often in case of rare words but corresponding word pairs should have similar frequencies, hence the feature indicating their difference. Features l_s and l_t supply further surface information about the words, while similarity and rank features are indicative of the quality of the candidates. We feed BWEs and BOEs of the word pair along with the additional features to the final feed-forward classifier.

We train the classifier to minimize the binary cross-entropy loss over positive (translation or transliteration pairs) and negative word pairs. The positive samples are the pairs in the training dictionary. We generate two negative samples for each source word: we take one candidate each from the two sorted lists of candidates (from BWEs and from seq2seqTr) at random between the 10th and 20th position, assuming that no positive pairs or their close synonyms belong to this range, but the words are still similar to the false candidates tested during prediction. We experimented with the range between 10 and 100, but the performance dropped since many words were used as negatives that are not realistic candidates for prediction. We note that a similar approach was developed in contemporary work (Karan et al., 2020).

3.5 Discussion

As discussed above the ensembling approach of Severini et al. (2020) has a limited reranking capability which is shown by our results. In contrast, our system is able to consider multiple candidates from the BWEs and from the transliteration candidate lists and re-rank them given the supplied information. For example, consider a non-transliteration pair such as *smoking*→курение for which a false friend exists: смокинг (*tuxedo*). The classifier can rank the false friend low since the frequencies of source and target words do not match. Although the system of Heyman et al. (2017) is based on a classifier, similar to our approach, it fails to pick the gold translation when it is not one of the top candidates in the BWEs

[2] Section 4 details minor modifications to the originally published system that make it suitable for our setup.

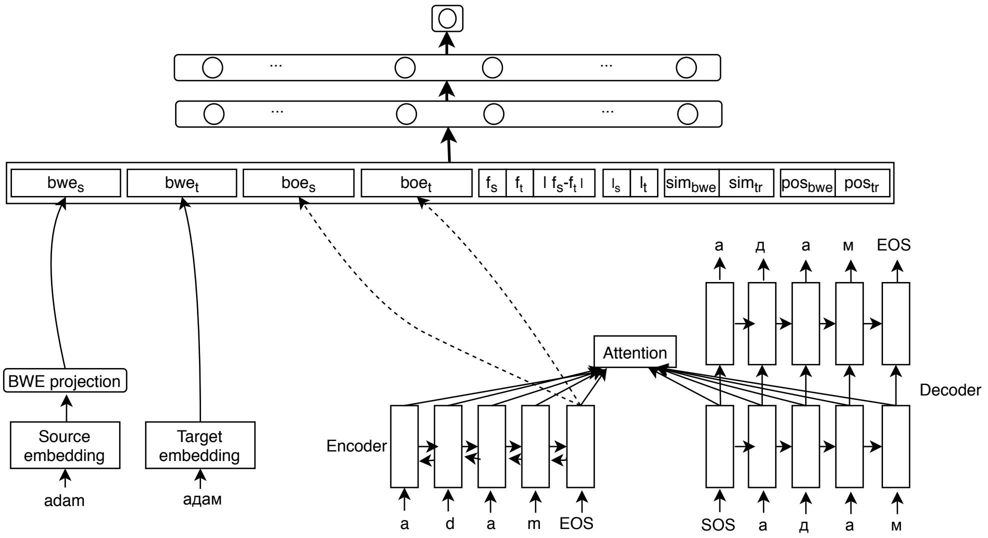


Figure 1: Classifier architecture. Top: input layer, two fully-connected layers and an output unit. Bottom right: the encoder-decoder model seq2seqTr, used as the transliteration model and to extract BOEs.

or transliteration candidates since it lacks the additional information coming from our proposed features. Statistics on the reranking capabilities of the models are shown below in section 5.1.

4 Experimental Setup

We ran our BDI experiments on the BUCC 2020 Shared Task dataset (Rapp et al., 2020); it provides both monolingual corpora and bilingual dictionaries for English-Russian. Since the official test set of the shared task is undisclosed, we relied on the released training set, a random subset of the MUSE dictionaries[3] released by Conneau et al. (2018). It is divided into three subsets, high, middle and low frequency, containing words ranked between 1-5000, 5001-20000 and 20001-40000 in the original dictionary, each having 2000 unique[4] source words. We split them into train, development and test sets (70%/15%/15%). We run experiments on the three frequency sets separately and jointly.

We followed the official setup of the BUCC shared task and relied on the WaCKy corpora (Baroni et al., 2009) as monolingual data to get the full language vocabularies and their frequencies. As MWEs we used the fasttext skipgram models (Bojanowski et al., 2017) released by the shared task organizers, which were trained using the WaCKy corpora with the following parameters: minimum word count 30; vector dimension 300; context window size 7; number of negatives sampled 10 and number of epochs 10. To align them we used VecMap (Artetxe et al., 2018) in a supervised setup using only high frequency word pairs for training, since less frequent words can be detrimental for mapping quality (Vulic and Korhonen, 2016). We use the BWEs of the 200K most frequent words following (Conneau et al., 2018). In addition, in an ablation study described below we used the 1000 word pairs of the NEWS 2010 transliteration mining test set (Kumaran et al., 2010) to test the quality of the BOEs.

For seq2seqTr we use learning rate 0.01, batch size 32, hidden size 128, single encoder and single decoder layers with the “dot” attention method of Luong et al. (2015) that compares the states with their dot product score. As mentioned above we use the same architecture for building BOEs, but we use different parameters. We tuned the hidden size (2000) of the GRU layers (used as the BOEs) and the character embedding size (300) along with the classifier on the BUCC high, middle and low frequency dev sets jointly.

For training the BDI classifier we used Adam optimizer (Kingma and Ba, 2014) with learning rate 0.001, batch size 32, 2 hidden layers on top of the described features with hidden size 300, dropout 0.01

[3] Contains up to 100K word pairs, translated with a MT system.

[4] Some words have multiple translation options.

	All	**High**	**Mid**	**Low**
BWEs with CSLS	0.29 (0.46)	0.47 (0.73)	0.29 (0.45)	0.11 (0.21)
Transliteration	0.05 (0.10)	0.05 (0.10)	0.06 (0.09)	0.05 (0.11)
(Severini et al., 2020)	0.33 (0.52)	0.50 (0.76)	**0.33** (0.51)	0.16 (0.30)
(Heyman et al., 2017)	0.21 (0.39)	0.28 (0.52)	0.22 (0.40)	0.14 (0.25)
(Heyman et al., 2017) w/ features	0.30 (0.48)	0.47 (0.72)	0.29 (0.45)	0.15 (0.25)
Proposed w/o features	0.22 (0.38)	0.33 (0.55)	0.23 (0.36)	0.12 (0.24)
Proposed w/o BOEs	0.31 (0.48)	0.50 (0.75)	0.30 (0.47)	0.12 (0.22)
Proposed	**0.36 (0.55)**	**0.55 (0.76)**	**0.33 (0.56)**	**0.19 (0.33)**

Table 1: $acc@1$ ($acc@5$) on the test set for All (High+Mid+Low), High, Mid and Low. The first two lines are obtained by taking the top-1 or top-5 (in brackets) highest scoring candidates from BWE candidates or seq2seqTr candidates. For $acc@1$, Severini et al. (2020) rank $m = 100$ candidates while Heyman et al. (2017) and we rank $m = 2$ and $m = 4$, respectively. For $acc@5$, Severini et al. (2020) rank $m = 100$ candidates while Heyman et al. (2017) and we rank $m = 5$.

and ReLu activation function on the inner layer. We used early stopping on the joint dev set, decreasing the learning rate by 0.1 after not improving for 10 steps. The encoder model is kept frozen during the classifier training.

All our models are implemented in Python using PyTorch (Paszke et al., 2019), including the re-implementation of (Heyman et al., 2017). We had to modify the original setup of Heyman et al. (2017) since it relied on Levenshtein distance to look for translation candidates for source words, which is not applicable in case of language pairs with different scripts; we instead use the same transliteration candidates as in our proposed system. Furthermore, we use BWEs as the word representations as opposed to the original work where MWEs were used and the system had to learn the alignment.

5 Experiments and Results

In this section, we show the main results of our approach and compare them with the baselines. We also show an evaluation of the BOEs on the NEWS 2010 shared task (Kumaran et al., 2010) to better understand their quality on a dataset that only contains transliteration pairs.

Table 1 shows the main results for our BDI system. Our evaluation measure is $acc@n$ ($n \in \{1, 5\}$): we take n predictions from a given model and consider the source word correctly translated if any of the n predictions is the gold translation. Other than the two baseline systems we show BDI performance by taking the n highest scoring words as predictions from only BWEs or only transliteration candidates. We tuned the parameters of all systems on the joint development set, except that we followed the approach of Severini et al. (2020) and tuned ensembling weights on the three frequency sets separately. Since the systems of Severini et al. (2020), Heyman et al. (2017) and our approach are able to re-rank translation candidates, we tune the number of candidates (m) considered during prediction on the development set, i.e., we take the m highest scoring words from both BWEs and seq2seqTr lists, re-score them using one of the mentioned systems and consider the first n as translations. $m = 100$ works best on the development set for (Severini et al., 2020), $m = 2$ for (Heyman et al., 2017), $m = 1$ works best for (Heyman et al., 2017) with features, and $m = 4$ works best for our system with $acc@1$. When measuring $acc@5$, $m = 5$ works best for the classifiers and $m = 100$ for (Severini et al., 2020). Larger m values lead to worse predictions due to noisy elements in the candidate lists in case of the classifiers. In contrast, (Severini et al., 2020) is more robust against noisy elements since it is only able to rerank if a candidate word is contained in both lists. Given that the BUCC test set is divided into frequency subsets, we analyze the performance also for those.

Table 1 shows that our approach outperforms all previous approaches both on the joint ("All") and the separate frequency sets. The BWE based approach in the first row of the table achieves high performance on the high frequency set, but it suffers a significant drop as word frequency decreases. The transliteration model by itself managed to correctly induce some of the words achieving around 10% $acc@5$, which

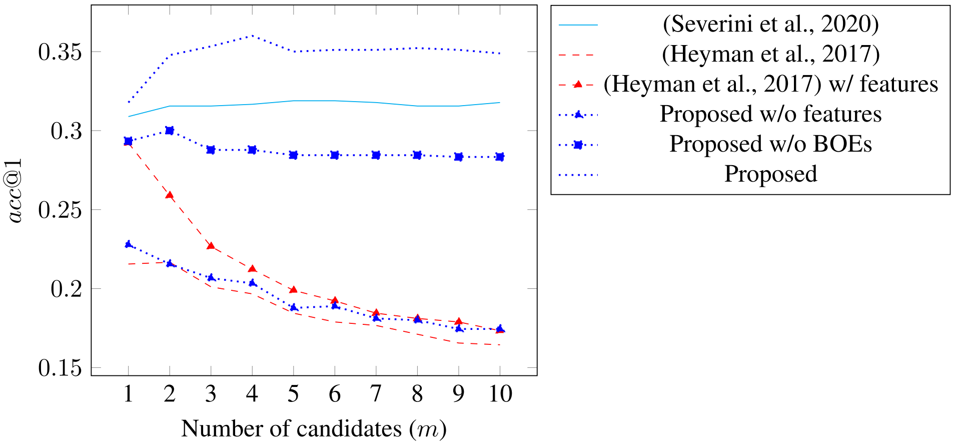


Figure 2: $acc@1$ on the development set as a function of the number of candidate words (e.g., 2 means 2 candidates from BWEs and 2 from seq2seqTr).

shows that a significant amount of words have to be transliterated in the datasets. The ensembling approach of Severini et al. (2020) combines the two sources of information well – the $acc@1$ score is almost the sum of the two combined models. It also outperforms the classifier baseline of Heyman et al. (2017). As mentioned above, best results were achieved with the classifier baseline when considering only 1 or 2 candidates each from BWEs and seq2seqTr, indicating that the system is struggling with reranking candidates. In contrast, our approach is able to exploit the additional candidates and achieves best results on all frequency sets in terms of $acc@5$ and three out of four sets in terms of $acc@1$.

5.1 Reranking Analysis

To analyze the reranking capability of our model, Figure 2 shows the performance as a function of the number of candidate words (m) on the dev set. The performance of our model improves as the number of candidates increases up to $m = 4$, and after a small performance drop at $m = 5$, it has a stable performance until 10 candidates, which further emphasizes that it is able to re-rank the candidates. When no features are used for our model, the best result is with one candidate and the performance decreases together with the re-ranking capability when using more candidates, indicating that the features are relevant to the system. Finally, we can see the behavior of our model when we rely on word embeddings and features but not on BOE information. The model acts similarly to the full version improving the results by using a few more candidates but performing near constantly after a drop as the number of candidates gets larger. On the other hand, the performance is constantly lower than the results of the full system meaning that the BOEs play a crucial role and are able to encode the orthographic structure of the words. We also added our novel features to the classifier of Heyman et al. (2017) to show the performance gain by themselves. Based on Figure 2 and Table 1, we can conclude that features clearly improve the performance of the baseline system, meaning that they are useful to decide if the BWEs or the character-level information should be emphasized. On the other hand, they are not enough to be able to positively exploit more translation candidates than 1 from this model. Similarly to our model, the performance of Severini et al. (2020) improves constantly as m increases and plateaus around $m = 100$. Still, our approach achieves better performance overall, since it does not require a given translation candidate to be contained in both candidate lists to be reranked. On the other hand, the performance of the classification based models drop significantly when $m > 10$ due to the noisier candidate lists while (Severini et al., 2020) is more robust against such noise since the noisy elements of the BWE and transliteration candidate lists are non-overlapping most of the time, thus they are not getting reranked. Our approach achieves 29.67% $acc@1$ with $m = 100$ but we only show $m \leq 10$ in figure 2 for simplicity.

We computed statistics, on the dev set, on the number of cases where a candidate is correctly chosen by the model and it is at rank 1 in neither the BWE nor the seq2seqTr ranking. We analyze the cases

	P	R	F
(Jiampojamarn et al., 2010)	88.0	86.9	87.5
(El-Kahky et al., 2011)	92.1	92.5	92.3
(Nabende, 2011)	-	-	82.5
(Sajjad et al., 2017)	67.1	97.1	79.4
BOEs	47.0	87.2	61.1
BOEs best	88.8	68.2	77.1

Table 2: Precision, Recall and F-measure for our BOEs and for state-of-the-art models on transliteration mining. (Sajjad et al., 2017) and our system are unsupervised while the others are (semi-) supervised.

where the best m value on the dev is greater than 1. Our model with $m = 4$ correctly predicts 15.7% of the candidates that are not in the first place while (Heyman et al., 2017) with $m = 2$ predicts only 8.8% and (Severini et al., 2020) with $m = 100$ predicts only 3.6%. Our model without BOEs is able to find only 3.8% of the candidates; thus, BOEs play a crucial role for the reordering capability of our system. We also consider the cases where the correctly chosen candidates are not at rank 1 against the total number of correct pairs found. 29.3% of the candidates correctly found by our model are not first ranked according to BWE and seq2seqTr, while 27.1% is the corresponding percentage for (Heyman et al., 2017). Note that the number of correct pairs are different for the two models and our proposed model translated more words correctly. Keeping this in mind, the small difference between the two models indicates that our model does not only have a performance advantage because of the better reranking capability, but it is also better at deciding to choose the first candidate from either the BWE or seq2seqTr lists.

5.2 BOE Evaluation

As shown above, BOEs are a crucial part of our classification model because they encode the similarity of two words based on their orthographic structure and not on their semantic meaning. The model with 2000 hidden BOE size and 300 character embedding size worked best on the BDI development set. To check the quality of the BOEs in a task where they are the only source of information, we conduct an evaluation on transliteration mining using the NEWS 2010 En-Ru test set (Kumaran et al., 2010). The task consists of the development of a mining system for identifying single word transliteration pairs from the Wikipedia Inter-Language Links (WIL) dataset in one or more language pairs. In particular, participants are required to identify word pairs in parallel sentences that are transliterations of each other. Note that the organizers provided a seed dataset of 1K transliteration pairs and a noisy training set which we ignore in these experiments and use the BUCC training dictionaries as already described to show the quality of BOEs which were used for the BDI task.

We pre-process the test sentences similar to (Kumaran et al., 2010). Given a pair of parallel sentences, for each word in the first sentence we look for the word in the second that has the highest score according to the model. We used the same model to obtain BOEs of words as in the BDI classifier. To get the score of two words, we calculate the cosine similarity of their BOEs, and we used a threshold of 0.5 to discriminate between pairs that are transliterations vs. those that are not (Sajjad et al., 2017). In Table 2 we show the precision, recall and f-measure for state-of-the-art models and our BOEs on this task. Our BOEs together with (Sajjad et al., 2017) are unsupervised and, although the BOEs are not specific for this task, they perform well. The last row of Table 2 shows the results when a threshold of 0.7 is used, that is, the best performing threshold found on the test set, thus it can be viewed as an oracle experiment. With this more accurate parameter, the system was able to reach better results compared to the naive threshold selection, which indicates the need for a development set. On the other hand, in this ablation study our goal was not to develop the best transliteration mining approach but to show the quality of BOEs. The good performance shows that BOEs have a universal embedding property of representing English and Russian words in a shared space although they use different scripts.

6 Conclusion

Bilingual Dictionary Induction is a relevant task for many applications and it is an important building block in the area of MT. In this paper we described our system for BDI for language pairs with different scripts focusing on words for which semantic information alone is insufficient. We combined semantic and orthographic information via transliteration. Our proposed model has the novel ability to make a reasonable decision on which source of information to choose via a classification approach that exploits – together with manually designed features – word embeddings and character-level information (BOEs). Our novel BOEs were learned by a language agnostic transliteration system. We tested our system on the English-Russian BUCC 2020 dataset and we showed improved results compared to the baselines. We also showed that our model is able to re-rank the candidate words better in contrast to other approaches. We evaluated our system on high, middle and low frequency sets separately and jointly. Finally, we evaluated our BOEs by running transliteration mining on the NEWS 2010 dataset, showing that they achieve good performance even if they were not meant for that specific task. Also, BOEs were shown to be able to encode the orthographic structure of words independent of the language. That means that they are universal embeddings that represent transliteration pairs similarly – a property that is useful for other downstream tasks as well. All in all, we presented a system able to combine word-level information with character-level information by means of transliteration and classification models for BDI that improved the baseline results.

Acknowledgements

We gratefully acknowledge funding for this work by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreements № 640550 and № 740516). The work was also supported by the German Research Foundation (DFG; grant FR 2829/4-1).

References

Mikel Artetxe and Holger Schwenk. 2019. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203.

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

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. 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.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An Effective Approach to Unsupervised Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

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.

Fabienne Braune, Viktor Hangya, Tobias Eder, and Alexander Fraser. 2018. Evaluating bilingual word embeddings on the long tail. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 188–193.

Nancy Chen, Rafael E Banchs, Min Zhang, Xiangyu Duan, and Haizhou Li. 2018. Report of news 2018 named entity transliteration shared task. In *Proceedings of the seventh named entities workshop*, pages 55–73.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word Translation Without Parallel Data. In *Proceedings of the International Conference on Learning Representations*, pages 1–14.

Paula Czarnowska, Sebastian Ruder, Edouard Grave, Ryan Cotterell, and Ann Copestake. 2019. Don't Forget the Long Tail! A Comprehensive Analysis of Morphological Generalization in Bilingual Lexicon Induction. 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)*, pages 974–983.

Ali El-Kahky, Kareem Darwish, Ahmed Saad Aldein, Mohamed Abd El-Wahab, Ahmed Hefny, and Waleed Ammar. 2011. Improved transliteration mining using graph reinforcement. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1384–1393.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: Hlt*, pages 771–779.

Geert Heyman, Ivan Vulić, and Marie Francine Moens. 2017. Bilingual lexicon induction by learning to combine word-level and character-level representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1085–1095.

Ann Irvine and Chris Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim, and Grzegorz Kondrak. 2010. Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, pages 39–47.

Mladen Karan, Ivan Vulić, Anna Korhonen, and Goran Glavaš. 2020. Classification-Based Self-Learning for Weakly Supervised Bilingual Lexicon Induction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6915–6922.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*, pages 9–16.

A Kumaran, Mitesh M Khapra, and Haizhou Li. 2010. Whitepaper of news 2010 shared task on transliteration mining. In *Proceedings of the 2010 Named Entities Workshop*, pages 29–38.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-Based & Neural Unsupervised Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of news 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 1–18.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting Similarities among Languages for Machine Translation. *CoRR*, abs/1309.4.

Peter Nabende. 2011. Mining transliterations from wikipedia using dynamic bayesian networks. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 385–391.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Reinhard Rapp, Pierre Zweigenbaum, and Serge Sharoff. 2020. Overview of the Forth BUCC Shared Task: Bilingual Dictionary Induction from Comparable Corpora. In *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 1–6.

Parker Riley and Daniel Gildea. 2018. Orthographic Features for Bilingual Lexicon Induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 390–394.

Hassan Sajjad, Helmut Schmid, Alexander Fraser, and Hinrich Schütze. 2017. Statistical models for unsupervised, semi-supervised, and supervised transliteration mining. *Computational Linguistics*, 43(2):349–375.

Silvia Severini, Viktor Hangya, Alexander Fraser, and Hinrich Schütze. 2020. LMU Bilingual Dictionary Induction System with Word Surface Similarity Scores for BUCC 2020. In *Proceedings ofthe 13th Workshop on Building and Using Comparable Corpora*, pages 49–55.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ivan Vulic and Anna Korhonen. 2016. On the Role of Seed Lexicons in Learning Bilingual Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 247–257.

Chapter 4

Corresponds to the following publication:

> **Silvia Severini**, Viktor Hangya, Masoud Jalili Sabet, Alexander Fraser, and Hinrich Schütze. "Don't Forget Cheap Training Signals Before Building Unsupervised Bilingual Word Embeddings.". In Proceedings of the 15th Workshop on Building and Using Comparable Corpora. 2022.

Resource: `https://github.com/silviaseverini/cheap-signals-for-BWEs.git`

Declaration of Co-Authorship: I conceived of the original contribution of considering efficient and effective signals for Bilingual Dictionary Induction and exploiting romanization. I designed the algorithms and performed all the experiments. Viktor Hangya ran state-of-the-art models and contributed with regular discussions. I wrote the initial draft of the article and did most of the subsequent corrections. All authors helped review the final draft of the paper and gave advice.

Proceedings of the 15th Workshop on Building and Using Comparable Corpora (BUCC 2022) @LREC2022 , pages 15–22
Marseille, June 25, 2022
© European Language Resources Association (ELRA), licensed under CC-BY-NC

Don't Forget Cheap Training Signals Before Building Unsupervised Bilingual Word Embeddings

Silvia Severini, Viktor Hangya, Masoud Jalili Sabet, Alexander Fraser, Hinrich Schütze
Center for Information and Language Processing
LMU Munich, Germany
{silvia, hangyav, masoud, fraser}@cis.uni-muenchen.de

Abstract

Bilingual Word Embeddings (BWEs) are one of the cornerstones of cross-lingual transfer of NLP models. They can be built using only monolingual corpora without supervision leading to numerous works focusing on unsupervised BWEs. However, most of the current approaches to build unsupervised BWEs do not compare their results with methods based on easy-to-access cross-lingual signals. In this paper, we argue that such signals should always be considered when developing unsupervised BWE methods. The two approaches we find most effective are: 1) using identical words as seed lexicons (which unsupervised approaches incorrectly assume are not available for orthographically distinct language pairs) and 2) combining such lexicons with pairs extracted by matching romanized versions of words with an edit distance threshold. We experiment on thirteen non-Latin languages (and English) and show that such cheap signals work well and that they outperform using more complex unsupervised methods on distant language pairs such as Chinese, Japanese, Kannada, Tamil, and Thai. In addition, they are even competitive with the use of high-quality lexicons in supervised approaches. Our results show that these training signals should not be neglected when building BWEs, even for distant languages.

Keywords: Bilingual Word Embeddings, Bilingual Dictionary Induction, Romanization

1. Introduction

Bilingual Word Embeddings (BWEs) are useful for many cross-lingual tasks. They can be built effectively even when only a small seed lexicon is available by mapping monolingual embeddings into a shared space. This makes them particularly valuable for low-resource settings (Mikolov et al., 2013). In addition, unsupervised mapping approaches can build BWEs for some languages when no seed lexicon is available. Various unsupervised methods have been proposed relying on the assumption that embedding spaces are isomorphic (Zhang et al., 2017; Lample et al., 2018; Artetxe et al., 2018; Alvarez-Melis and Jaakkola, 2018; Chen and Cardie, 2018; Hoshen and Wolf, 2018; Mohiuddin and Joty, 2019; Alaux et al., 2019; Dou et al., 2020; Grave et al., 2019; Li et al., 2020). However, with one exception, none of them compare their results with the widely available baseline of using identical words as seed lexicons.

It has been shown that identical word pairs of two languages can be used to build high quality BWEs (Smith et al., 2017; Artetxe et al., 2017). However, they were only tested on language pairs with similar scripts. The only exception is the work of Søgaard et al. (2018), who tested identical word pairs on English and Greek which use different alphabetical characters but the same numerals. Regardless of these experiments, recent works still propose novel unsupervised approaches without considering such cheap training signals, at least as baseline systems (Mohiuddin and Joty, 2019; Alaux et al., 2019; Dou et al., 2020; Grave et al., 2019; Li et al., 2020).

In this paper however, we argue that such signals should be used as a cheap and effective baseline in the development of future unsupervised methods. We define them cheap as they require widely available monolingual corpora only, e.g., Wikipedia dumps, but no parallel data. We study two approaches for extracting the initial seed lexicons to build BWEs without relying on expensive dictionaries. (1) First, we leverage identical pairs as proposed by Smith et al. (2017; Artetxe et al. (2017). Previous work assumed such pairs not to be available for language pairs with distinct scripts, hence the development of various unsupervised mapping approaches. We show that, surprisingly, they do appear in large quantities in the monolingual corpora that we use, even for distinct-script pairs. In contrast to Søgaard et al. (2018), we test identical word pairs on multiple language pairs with distinct scripts, including pairs using distinct numerals. In addition, we propose to (2) strengthen identical pairs by extending them with further easily accessible pairs based on romanization and edit distance, which exploits implicit links between languages in the form of approximate word transliteration pairs.

We focus on distant language pairs having distinct scripts for many of which unsupervised approaches have failed or had very poor performance so far. For instance, English to Chinese, Japanese, Kannada, Tamil, and Thai, which all obtain a score close to 0 on the Bilingual Dictionary Induction (BDI) task (Vulić et al., 2019). We evaluate the two approaches on thirteen different non-Latin[1] languages paired with English on BDI. We compare our lexicons' performance with unsupervised mapping and the frequently used MUSE training lexi-

[1] We use *(non-)Latin language* here as a short form for *language standardly written in a (non-)Latin script*.

cons (Lample et al., 2018) and show that our noisy word pairs make it possible to build BWEs for language pairs where unsupervised approaches failed before and give accuracy scores similar to high quality lexicons.

Our work calls into question – at least for BDI – the strong trend toward unsupervised approaches in recent literature, similarly to Vulić et al. (2019), given that cheap signals are (i) available and easy to exploit, (ii) sufficient to obtain performance similar to dictionaries based on parallel resources like MUSE and (iii) able to make up for the failure of unsupervised methods. Finally, we analyze which lexicon properties impact performance and show that our lexicon outperform unsupervised methods also for non-English language pairs. Our paper calls for the need to use easily accessible bilingual signals, such as identical and/or transliteration word pairs, as baselines when developing unsupervised BWE approaches.

2. Unsupervised pair extraction

We show that we can extract the seed lexicon needed for mapping systems without the need for labeled data, making up for the failure of unsupervised methods. First, we show that identical pairs do appear in corpora of distant languages and can be exploited. Secondly, we propose a novel method to boost the identical pairs sets by extracting the initial seed lexicon without the need for any bilingual knowledge, starting from monolingual corpora, and using romanization and edit distance.

2.1. Identical pair approach

When dealing with languages with different scripts, identical pairs would seem to be unlikely to occur, which is assumed by unsupervised mapping methods. Smith et al. (2017; Artetxe et al. (2017) form dictionaries from identical strings which appear in both languages but limit their approach to similar languages sharing a common alphabet, such as European ones. Similarly, (Lample et al., 2018) refrain from using such identical word pairs, assuming they are not available for distant languages. An exception is the work of Søgaard et al. (2018) which shows the presence of identical pairs between English and Greek, which share numerals only but not alphabetical characters.

However, we show that there are domains where these pairs are actually available in large quantity even for pairs with different scripts, including the use of different numerals; an example is Wikipedia: see the statistics of fastText Wikipedia embeddings (Bojanowski et al., 2017) in Table 1. Most of these identical pairs are punctuation marks and digits, non-transliterated named entities written in the Latin script, or English words (assumingly words of a title) which were not translated in the non-English languages. This is also true for language pairs not including English. In this paper, we build BWEs based on these pairs and show that they are sufficient for good BDI results on distant language pairs with distinct scripts.

Lang	ID	Lang	ID	Lang	ID
ko-th*	17K	ko-he*	11K	he-th*	15K
en-zh*	62K	en-bn*	31K	en-ar*	19K
en-th	46K	en-hi*	30K	en-ru	18K
en-ja	43K	en-ta*	23K	en-he*	17K
en-el	35K	en-kn*	21K	en-ko*	15K
en-fa*	32K				

Table 1: Number of identical pairs per language pair. Language pairs using different digits as their official numerals, on top of different alphabetical characters, are indicated with *.

2.2. Romanization based augmentation (ID++)

Identical pairs are noisy and may appear in smaller quantities for certain corpora and language pairs (e.g., he-ko). We propose our romanization approach that builds the seed lexicon completely automatically and can augment the identical pairs set. We exploit the concept of transliteration and orthographic similarity to find a cheap signal between languages (cf. (Riley and Gildea, 2018; Severini et al., 2020a; Severini et al., 2020b; Severini et al., 2022)) and to take advantage of cognates (Chakravarthi et al., 2019; Laville et al., 2020). It consists of 3 steps at the end of which we add the identical pairs and run VecMap in a semi-supervised setting.

1. Source candidates First, we generate a list of source language words, which are the candidates to be matched with a word on the target side. We use the English Wikipedia dumps[2] as our monolingual corpus and apply Flair (Akbik et al., 2018) to extract Universal Part-of-Speech (UPOS) tags. We collect all English proper nouns (PROPN), since names are often transliterated between languages. The resulting English proper noun set consists of ≈800K words.

2. Target candidates The language-specific target data is extracted from the vocabulary of the pre-trained Wikipedia fastText embeddings (Bojanowski et al., 2017). The sets are not pre-processed with a POS tagger assuming that such a tool is missing or perform poorly for low-resource languages. Compared to the English proper noun set, the vocabularies are smaller: between 40K and 500K. Then, we romanize the corpora to obtain equivalent words but with only Latin characters – this supports the distance-based metrics in step (3). We use Uroman (Hermjakob et al., 2018) for romanization. Examples of romanization are карл (Russian)→ carl and βαβυλών (Greek) → babylon. Uroman mainly covers 1-1 character correspondences and does not vocalize words for Arabic and Hebrew. In general, its romanization is not as accurate as the transliteration of a neural model. However, neural models need a training corpus of labeled pairs to work well, while Uroman only

[2] https://dumps.wikimedia.org/ (01.04.2020)

	en-th	en-ja	en-kn	en-ta	en-zh
Unsupervised					
1.	0.00	0.96	0.00	0.07	0.07
2.	0.00	0.48	0.00	0.07	0.00
3.	0.00	0.00	0.00	0.00$^\diamond$	0.00
Semi-supervised (Artetxe et al., 2018)					
ID	24.40	48.87	22.03	17.93	37.00
Rom.	23.33	48.46	22.90	18.00	0.27
ID++	23.47	49.14	24.23	18.20	35.00
MUSE	24.33	48.73	23.78	18.80	36.53

Table 2: acc@1 on BDI for unsupervised (1: Artetxe et al. (2018), 2: Grave et al. (2019), 3: Mohiuddin and Joty (2019)) and semi-supervised approaches for 5 languages for which unsupervised methods fail. The semi-supervised results are obtained using VecMap with three different initial lexicons: the identical pair set (ID), ID extended with romanization based pairs (ID++) and the MUSE dictionary. We show an ablation study as well, i.e., the romanized pairs only (Rom.). Scores from Mohiuddin et al. (2020) are marked with $^\diamond$.

uses the character descriptions from the Unicode table,[3] manually created tables and some heuristics, supporting a large number of languages.

3. Candidate matching To find the corresponding target word for an English noun, the noun is compared with each (romanized) target word based on their orthography. The similarity of two words w_1 and w_2 is defined as $1 - \text{NL}(w_1, w_2)$, where NL is the Levenshtein distance (Levenshtein, 1966) divided by the length of the longer string. We select a pair of words if the similarity is ≥ 0.8; this ensures a trade off between number of pairs and quality, based on manual investigation. We use the Symmetric Delete algorithm to speed up computation, similarly to (Riley and Gildea, 2018). It takes the lists of source and target words, and a constant k and identifies all the source-target pairs that are identical after k insertion or deletions.[4] The final step is to look up, for each romanized target word, its original non-romanized form.

3. Evaluation

We evaluate our seed lexicons on BDI to show the quality of the BWEs obtained with them. Recent papers (Marchisio et al., 2020) show that there is a direct relationship between BDI accuracy and downstream BLEU for machine translation. Moreover, Sabet et al. (2020) show that good-quality word embeddings directly reflect the performance also for extrinsic tasks like word alignment. We use the VecMap tool to build BWEs since it supports both unsupervised, semi-supervised and supervised techniques (Artetxe et al., 2018). The semi-supervised approach is of particular interest to us since it performs well with small and noisy seed lexicons by iteratively refining them. VecMap iterates over two steps: embedding mapping and dictionary induction. The process starts from an initial dictionary that is iteratively augmented and refined by extracting probable word pairs from the BWEs built in the current iteration with BDI. The method is repeated until the improvement on the average dot product for the induced dictionary stays above a given threshold. We use pretrained Wikipedia fastText embeddings (Bojanowski et al., 2017) as the input monolingual vectors, taking only the 200K most frequent words and using default parameters otherwise. We compare the performance of VecMap using our lexicons with MUSE. MUSE contains dictionaries for many languages and it was created using a Facebook internal translation tool (Lample et al., 2018), thus it can be considered as a higher quality cross-lingual resource based on parallel data. Since Kannada is not supported by MUSE, we use the dictionary provided by Anzer et al. (2020). We show $acc@1$ scores based on CSLS vector similarity calculated by the MUSE evaluation tool (Lample et al., 2018).[5]
Tables 2 and 3 show accuracy for all language pairs considering English as the source; see Table 7 in Appendix B for the full table containing results in both directions. Table 2 gives scores for language pairs for which unsupervised methods completely diverge (acc@1 < 1). We report results for three unsupervised methods (Artetxe et al., 2018; Mohiuddin and Joty, 2019; Grave et al., 2019). In contrast, using identical word pairs as lexicon (ID) or its extension with the romanizetion based pairs (ID++) with VecMap leads to successful BWEs without any parallel data or manually created lexicons. In addition, scores are even comparable to high-quality dictionaries like MUSE. Looking at results for all language pairs in Table 2 and 3, our sets always obtain results comparable to MUSE (baseline dictionaries), with improvements for Arabic, Chinese, Russian and Greek. In the unsupervised cases (Table 2), both ID and ID++ pair sets lead to an accuracy improvement of at least 17 points. ID++ outperform ID for three of the five low-resource pairs and five out of eight high-resource pairs proving that the romanized pairs can indeed strengthen the identical pairs sets. These results show that good quality BWEs can be built by relying on implicit cross-lingual signals without expensive supervision or fragile unsupervised approaches.

MUSE test w/o proper nouns The work of Kementchedjhieva et al. (2019) highlights that MUSE test sets contain a high number of proper nouns for German, Danish, Bulgarian, Arabic and Hindi. Since our romanization augmentation is based on such names, we evaluate their performance on the subsets of MUSE test

[3] http://unicode.org/Public/UNIDATA/UnicodeData.txt

[4] We used minimum frequency and minimum length equal to 1, k equals to 2.

[5] We follow Artetxe et al. (2018) work for comparison reasons and did not remove identical pairs from the test sets. However, overlaps between train romanized lexicons and test lexicons correspond to less than 1%.

	Unsup.	ID	Rom.	ID++	MUSE
en-ar	36.30	40.27	39.33	40.20	39.87
en-hi	40.20	40.47	39.60	40.20	40.33
en-ru	44.80	49.13	48.87	49.53	48.80
en-el	47.90	47.87	48.00	48.27	48.00
en-fa	36.70	37.67	36.80	37.67	38.00
en-he	44.60	44.47	44.53	44.67	45.00
en-bn	18.20	19.87	19.80	20.13	21.60
en-ko	19.80	27.92	28.40	28.81	28.94

Table 3: acc@1 on BDI for (best) unsupervised method and semi-supervised VecMap with different initial lexicons. (full table in Appendix B, Table 7).

	Unsup.	ID	Rom.	ID++	PanLex
th-ko	0.00	2.81	3.37	3.09	2.95
th-he	0.00	9.75	0.00	8.86	10.13
ko-th	0.00	15.90	14.23	15.26	14.36
ko-he	14.62	15.68	16.08	16.00	15.11
he-th	0.00	16.42	0.00	16.54	17.90
he-ko	14.30	15.39	15.15	15.09	16.06

Table 4: acc@1 on BDI for unsupervised and semi-supervised VecMap for all combinations of Korean, Hebrew, and Thai. PanLex are results obtained with training lexicons from Vulić et al. (2019) and semi-supervised VecMap.

sets that don't contain proper nouns. We remove proper nouns using the list of names obtained in Section 2.2 and evaluate the performance of all the approaches presented above. The new sets contains 10% less pairs on average. Results are shown in Table 8, Appendix C. The performance is similar to the one obtained on the original test sets, proving that our dictionaries and methods are not biased towards aligning word embeddings of proper nouns.

Non-English centric evaluation We analyze the performance of ID and ID++ for language pairs that do not include English. We use the test dictionaries from Vulić et al. (2019) that are derived from PanLex (Baldwin et al., 2010; Kamholz et al., 2014) by automatically translating each source language word into the target languages. We run VecMap for all combinations of Korean, Hebrew, and Thai. Romanized train lexicons are extracted by combining the languages through English (e.g., th-ko is obtained using en-th and en-ko), i.e., words are paired if their English translation is the same. Table 4 shows results. When Thai is involved, the unsupervised method fails as for English-Thai. Both ID and ID++ always outperform the respective unsupervised scores, and perform similar to higher-quality dictionaries. Additionally, ID++ outperforms ID in 3 out of 6 cases. These results demonstrate further the simplicity and high quality of our methods.

Romanized-only We analyze the performance of romanized pair lexicons on their own. Line Rom. in Table 2 and 3 shows that they obtain competitive results to the other two approaches, with improvements for Japanese, and perform similarly to MUSE dictionaries. The only failure is for Chinese (en-zh) – presumably because Chinese has a logographic script that does not represent phonemes directly, so romanization is less effective. These results show that the romanized pairs on their own also represent strong signals that shouldn't be neglected. Moreover, they constitute a good alternative when identical pairs are not available is such quantities (e.g., corpora of religious domain, law field, or cultural-specific documents).

Impact of OOVs We analyze the pairs used for the various sets (Appendix A, Table 5). We define OOVs as words for which there is no embedding available among the pre-trained Wikipedia fastText embeddings. Our romanized sets contain a substantial number of OOVs. (The identical pair sets do not contain OOVs because words are extracted from the top 200K most frequent.) The main reason for OOVs is that the selected English pair of a word is so rare that they do not have embeddings. On the other hand, the high number of OOVs (and resulting reduction of usable pairs) has only a limited negative impact on the performance.

Size of seed set and word frequency We analyze the impact of the size of the initial romanized seed set and of word frequency. Appendix A, Table 6, displays accuracy scores for MUSE and Romanized lexicons containing the $n \in \{25, 1000\}$ least and most frequent word pairs. Performance of VecMap applied to seed sets of size 25 is close to 0. The only exception is Russian, where the unsupervised approach already works well. Next, we investigate seed sets of size 1000 consisting of either the least frequent or the most frequent words. High-frequency seed sets give better results as expected. The effect is particularly strong for Tamil: the high-frequency set has performance close to the full set, whereas the low-frequency set is at $\leq$0.07. The performance of MUSE seed sets of size 25 and romanized seed sets of size 1000 is similar, demonstrating the higher quality of MUSE. However, obtaining the romanized pairs is much cheaper.

4. Conclusion

We have analyzed two cheap resources for building BWEs which can alleviate the issues of unsupervised methods which fail on multiple language pairs. We focused on a wide range of non-Latin languages paired with English. (i) We exploited identical pairs that surprisingly appear in corpora of distinct scripts. We showed that they can be used even when numerals are distinct in contrast to previous work. (ii) We combined them with a simple method to extract the initial hypothesis set via romanization and edit distance. With both approaches, we obtained results that are competitive with high-quality dictionaries. Without using explicit cross-lingual signal, we outperformed previous unsupervised work for most languages and in particular for five language pairs for which previous unsupervised work failed. Our results question the strong trend towards unsupervised mapping approaches, and show that cheap cross-lingual signals should always be considered for building BWEs, even for distant languages.

Acknowledgments

This work was funded by the European Research Council (grant #740516, #640550), the German Federal Ministry of Education and Research (BMBF, grant #01IS18036A), and the German Research Foundation (DFG; grant FR 2829/4-1).

5. Bibliographical References

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Alaux, J., Grave, E., Cuturi, M., and Joulin, A. (2019). Unsupervised hyperalignment for multilingual word embeddings. In *Proceedings of the 7th International Conference on Learning Representations*.

Alvarez-Melis, D. and Jaakkola, T. S. (2018). Gromov-wasserstein alignment of word embedding spaces. In *EMNLP*.

Anzer, M., Chronopoulou, A., and Fraser, A. (2020). Comparing unsupervised and supervised approaches for kannada/english bilingual word embeddings. *Bachelor thesis at Ludwig Maximilians Universität München*.

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

Artetxe, M., Labaka, G., and Agirre, E. (2018). 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.

Baldwin, T., Pool, J., and Colowick, S. (2010). Panlex and lextract: Translating all words of all languages of the world. In *Coling 2010: Demonstrations*, pages 37–40.

Chakravarthi, B. R., Arcan, M., and McCrae, J. P. (2019). Comparison of different orthographies for machine translation of under-resourced dravidian languages. In *2nd Conference on Language, Data and Knowledge (LDK 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Chen, X. and Cardie, C. (2018). Unsupervised multilingual word embeddings. *arXiv preprint arXiv:1808.08933*.

Dou, Z. Y., Zhou, Z. H., and Huang, S. (2020). Unsupervised bilingual lexicon induction via latent variable models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 621–626.

Grave, E., Joulin, A., and Berthet, Q. (2019). Unsupervised alignment of embeddings with wasserstein procrustes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1880–1890. PMLR.

Hoshen, Y. and Wolf, L. (2018). Non-adversarial unsupervised word translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 469–478.

Kamholz, D., Pool, J., and Colowick, S. (2014). Panlex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3145–3150.

Kementchedjhieva, Y., Hartmann, M., and Søgaard, A. (2019). Lost in evaluation: Misleading benchmarks for bilingual dictionary induction. *arXiv preprint arXiv:1909.05708*.

Laville, M., Hazem, A., and Morin, E. (2020). Taln/ls2n participation at the bucc shared task: bilingual dictionary induction from comparable corpora. In *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 56–60.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710.

Li, Y., Luo, Y., Lin, Y., Du, Q., Wang, H., Huang, S., Xiao, T., and Zhu, J. (2020). A simple and effective approach to robust unsupervised bilingual dictionary induction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5990–6001.

Marchisio, K., Duh, K., and Koehn, P. (2020). When does unsupervised machine translation work? *arXiv preprint arXiv:2004.05516*.

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

Mohiuddin, T. and Joty, S. (2019). Revisiting adversarial autoencoder for unsupervised word translation with cycle consistency and improved training. In *Proceedings of NAACL-HLT*, pages 3857–3867.

Mohiuddin, M. T., Bari, M. S., and Joty, S. (2020). Lnmap: Departures from isomorphic assumption in bilingual lexicon induction through non-linear mapping in latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2712–2723.

Riley, P. and Gildea, D. (2018). Orthographic features for bilingual lexicon induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 390–394.

Sabet, M. J., Dufter, P., Yvon, F., and Schütze, H. (2020). Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Severini, S., Hangya, V., Fraser, A., and Schütze, H. (2020a). Combining word embeddings with bilingual orthography embeddings for bilingual dictionary induction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6044–6055.

Severini, S., Hangya, V., Fraser, A., and Schütze, H. (2020b). Lmu bilingual dictionary induction system with word surface similarity scores for bucc 2020. In *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 49–55.

Severini, S., Imani, A., Dufter, P., and Schütze, H. (2022). Towards a broad coverage named entity resource: A data-efficient approach for many diverse languages. *arXiv preprint arXiv:2201.12219*.

Smith, S. L., Turban, D. H., Hamblin, S., and Hammerla, N. Y. (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*.

Søgaard, A., Ruder, S., and 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)*, pages 778–788.

Vulić, I., Glavaš, G., Reichart, R., and Korhonen, A. (2019). Do we really need fully unsupervised cross-lingual embeddings? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4398–4409.

Zhang, M., Liu, Y., Luan, H., and Sun, M. (2017). Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1970.

6. Language Resource References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Hermjakob, U., May, J., and Knight, K. (2018). Out-of-the-box universal romanization tool uroman. In *Proceedings of ACL 2018, System Demonstrations*, pages 13–18.

Lample, G., Conneau, A., Ranzato, M., Denoyer, L., and Jégou, H. (2018). Word translation without parallel data. In *International Conference on Learning Representations*.

A. Statistics

In this section we show statistics on the language pairs analyzed and additional scores. Table 5 presents the number of pairs for each set that are not OOVs in the fastText wiki embeddings (Bojanowski et al., 2017) .

	MUSE	ID	Romanized	ID++
en-th	6,799	46,653	10,721 / 53,804	58779 / 101066
en-ja	7,135	43,556	11,488 / 118,626	54970 / 161848
en-kn	1,552	21,090	12,888 / 59,207	33843 / 80032
en-ta	8,091	23,538	5,987 / 120,836	29472 / 143990
en-zh	8,728	62,289	6,360 / 41,829	68597 / 103971
en-ar	11,571	19,275	4,773 / 61,031	24019 / 80115
en-hi	8,704	30,502	16,180 / 73,553	46557 / 103791
en-ru	10,887	18,663	9,913 / 301,698	28520 / 319688
en-el	10,662	35,270	20,740 / 150,472	55841 / 185244
en-fa	8,869	32,866	10,226 / 85,210	43019 / 117817
en-he	9,634	17,012	4,005 / 40,258	20977 / 57059
en-bn	8,467	31,954	10,721 / 53,804	42573 / 85532
en-ko	7,999	15,518	9956 / 134156	25344 / 149031

Table 5: Number of pairs used that are not OOVs in the fastText wiki embeddings compared to the full size of the sets. For MUSE full and identical pairs sets there are no OOVs.

B. Main results

In Table 7 there are the accuracy scores based on CSLS vector similarity calculated by the MUSE evaluation tool (Lample et al., 2018). We show the scores for thirteen language pairs in both directions. The first five pairs are the ones for which unsupervised methods fail. We show both unsupervised and semi-supervised VecMap performance with baselines dictionaries and our three sets.

C. MUSE proper nouns removal

Table 8 shows results computed on the subsets of MUSE test sets that don't contain proper nouns. We remove proper nouns using the list of names obtained in Section 2.2 The new sets contains 10% less pairs on average.

D. Reproducibility

We run our method on up to 48 cores of Intel(R) Xeon(R) CPU E7-8857 v2 with 1TB memory and a single GeForce GTX 1080 GPU with 8GB memory. The training of semi-suprised BWEs using VecMap took approximately 1 hour per language pair. For VecMap, as well as for all others methods we analyzed, we used the latest code available in their git repositories with default parameters. ID++ is implemented in Python.

		MUSE				Rom.			
		25L	25H	1000L	1000H	25L	25H	1000L	1000H
en-ta	→	14.73	16.27	17.33	17.40	0.00	0.00	0.07	17.80
	←	16.48	18.35	22.44	23.44	0.00	0.00	0.00	21.57
en-fa	→	35.33	34.20	38.07	37.20	0.00	0.20	37.47	37.47
	←	41.73	42.60	44.14	44.21	0.07	0.13	42.40	43.40
en-zh	→	39.00	39.40	38.20	37.67	0.00	0.00	0.07	0.40
	←	32.93	34.47	34.33	34.40	0.00	0.00	0.07	0.60
en-ru	→	49.07	43.07	49.07	49.27	49.33	47.73	49.40	49.00
	←	65.93	60.60	65.93	66.13	65.80	64.47	65.60	66.40

Table 6: acc@1 using 25 or 1000 pairs lower-frequency (L) and higher-frequency (H) sets for MUSE and our romanized only (Rom.) set.

			Baselines				Our		
			Unsupervised			Semi-sup.	Semi-sup.		
			1	2	3	MUSE	ID	Rom.	ID++
1	en-th	→	0.00	0.00	0.00	**24.33**	**24.40**	23.33	23.47
		←	0.00	0.00	0.00	**19.04**	**19.92**	17.96	19.85
2	en-ja	→	0.96	0.48	0.00	**48.73**	48.87	48.46	**49.14**
		←	0.96	0.00	0.00	**32.87**	33.22	**34.80**	33.43
3	en-kn	→	0.00	0.00	0.00	**23.78***	22.03	22.90	**24.23**
		←	0.00	0.00	0.00	**41.25***	**43.04**	42.50	41.79
4	en-ta	→	0.07	0.07	0.00◇	**18.80**	17.93	18.00	**18.20**
		←	0.07	0.00	0.00◇	**24.38**	**24.78**	23.51	**24.78**
5	en-zh	→	0.07	0.00	0.00	**36.53**	**37.00**	0.27	35.00
		←	0.00	0.00	0.00	**32.80**	**34.33**	0.07	32.67
6	en-ar	→	33.60	7.67	36.30◇	**39.87**	**40.27**	39.33	40.20
		←	47.72	12.92	52.60◇	**54.48**	54.42	54.42	**54.62**
7	en-hi	→	40.20	0.00	0.00◇	**40.33**	**40.47**	39.60	40.20
		←	**50.57**	0.07	0.00◇	50.50	49.77	49.90	**50.10**
8	en-ru	→	**48.80**	37.33	46.90◇	**48.80**	49.13	48.87	**49.53**
		←	**66.13**	52.73	64.70◇	65.67	**66.13**	65.73	66.07
9	en-el	→	47.67	34.67	47.90◇	**48.00**	47.87	48.00	**48.27**
		←	63.40	49.20	**63.50◇**	63.33	63.27	**64.40**	63.47
10	en-fa	→	33.27	0.53	36.70◇	**38.00**	**37.67**	36.80	**37.67**
		←	39.99	0.40	**44.50◇**	43.47	**43.67**	42.93	43.60
11	en-he	→	44.60	37.13	44.00◇	**45.00**	44.47	44.53	**44.67**
		←	57.88	50.01	57.10◇	**57.94**	**58.14**	57.81	57.94
12	en-bn	→	18.20	0.00	0.00◇	**21.60**	19.87	19.80	**20.13**
		←	22.19	0.00	0.00◇	**28.46**	28.88	28.67	**29.41**
13	en-ko	→	19.80	9.62	0.00	**28.94**	27.92	28.40	**28.81**
		←	24.37	13.83	0.00	**34.09**	33.40	33.74	**33.95**

Table 7: acc@1 for unsupervised methods (1: Artetxe et al. (2018), 2: Grave et al. (2019), 3: Mohiuddin and Joty (2019)) and semi-supervised VecMap with different initial lexicons: MUSE set, identical pairs dataset (ID), our romanized only sets (Rom.), and the union of identical and romanized pairs (ID++). We show both forward (→) and backward (←) directions. In bold the best result for each pair of languages, for "Baselines" and "Our". Scores from Mohiuddin et al. (2020) are marked with ◇.
*Kannada is not supported by MUSE, so we use the dictionary provided by (Anzer et al., 2020).

			Baselines		Our		
			Unsup	Semi-sup.	Semi-supervised		
				MUSE	ID	Rom.	ID++
1	en-th	→	0.00	**27.21**	**27.13**	26.35	26.11
		←	0.00	18.93	19.83	18.25	19.83
2	en-ja	→	0.71	**46.15**	45.04	46.31	**46.39**
		←	0.56	**39.14**	38.86	**40.73**	39.52
3	en-kn	→	0.00	**23.78***	22.03	22.90	**24.23**
		←	0.00	**41.25***	**43.04**	42.50	41.79
4	en-ta	→	0.08	**20.12**	19.35	18.97	**19.43**
		←	0.08	**24.60**	24.60	23.71	**25.00**
5	en-zh	→	0.07	**37.34**	**38.14**	0.07	35.74
		←	0.00	**32.48**	**34.83**	0.00	32.48
6	en-ar	→	35.44	**39.70**	**40.23**	39.24	40.15
		←	49.75	**53.61**	53.46	53.61	**53.82**
7	en-hi	→	**42.49**	42.42	**42.79**	42.11	42.57
		←	52.46	**52.62**	51.99	52.07	**52.23**
8	en-ru	→	**45.64**	**45.64**	46.40	45.64	**46.70**
		←	**64.35**	64.13	64.57	64.35	**64.72**
9	en-el	→	48.90	**49.35**	48.97	49.43	**49.58**
		←	**63.87**	63.80	63.87	**64.56**	63.72
10	en-fa	→	34.18	**37.51**	37.35	36.58	**37.59**
		←	41.78	**43.59**	**44.06**	43.35	43.82
11	en-he	→	42.22	**42.60**	**42.29**	42.14	**42.29**
		←	**55.92**	55.70	56.00	55.62	**56.08**
12	en-bn	→	20.44	**22.74**	21.59	20.52	**20.98**
		←	25.80	**30.22**	30.30	30.30	**30.96**
13	en-ko	→	20.30	**26.57**	25.63	26.02	**26.49**
		←	26.52	**32.37**	**32.21**	31.80	32.13

Table 8: acc@1 on MUSE test sets without proper nouns. Results are reported for unsupervised and semi-supervised Vecmap Artetxe et al. (2018) with different initial lexicons: MUSE set, identical pairs dataset (ID), our romanized only sets (Rom.), and the union of identical and romanized pairs (ID++). We show both forward (→) and backward (←) directions. In bold the best result for each pair of languages, for "Baselines" and "Our".
*Kannada is not supported by MUSE, so we use the dictionary provided by (Anzer et al., 2020).

Chapter 5

Corresponds to the following publication:

> **Silvia Severini**, Ayyoob Imani, Philipp Dufter, and Hinrich Schütze. "Towards a Broad Coverage Named Entity Resource: A Data-Efficient Approach for Many Diverse Languages.". In Proceedings of the 13th Language Resources and Evaluation Conference. 2022.

Resource: `http://cistern.cis.lmu.de/ne_bible/`

Declaration of Co-Authorship: I conceived of the original research contribution. I designed the method and performed all implementations and evaluations except for sections 5.2 and 5.3 conceived by me but implemented by Ayyoob Imani. I regularly discussed this work with my advisor Hinrich Schütze. I wrote the initial draft of the article and did most of the subsequent corrections. All authors helped review the final draft of the paper.

Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 3923–3933
Marseille, 20-25 June 2022
© European Language Resources Association (ELRA), licensed under CC-BY-NC-4.0

Towards a Broad Coverage Named Entity Resource: A Data-Efficient Approach for Many Diverse Languages

Silvia Severini, Ayyoob Imani, Philipp Dufter*, Hinrich Schütze
Center for Information and Language Processing
LMU Munich, Germany
silvia@cis.uni-muenchen.de

Abstract

Parallel corpora are ideal for extracting a *multilingual named entity (MNE) resource*, i.e., a dataset of names translated into multiple languages. Prior work on extracting MNE datasets from parallel corpora required resources such as large monolingual corpora or word aligners that are unavailable or perform poorly for underresourced languages. We present CLC-BN, a new method for creating an MNE resource, and apply it to the Parallel Bible Corpus, a corpus of more than 1000 languages. CLC-BN learns a neural transliteration model from parallel-corpus statistics, without requiring any other bilingual resources, word aligners, or seed data. Experimental results show that CLC-BN clearly outperforms prior work. We release an MNE resource for 1340 languages and demonstrate its effectiveness in two downstream tasks: knowledge graph augmentation and bilingual lexicon induction.

Keywords: Low-resource,Multilinguality,Named Entities,Transliteration

1. Introduction

Of the thousands of languages in the world, a very small portion is covered by language technologies (Joshi et al., 2020). Bird (2020) suggests a number of approaches to develop such technologies for low-resource languages. In this paper, our goal is to create a *multilingual named entity (MNE) resource* – by which we mean a dataset of names translated into multiple languages – for a large number of low-resource languages, in total more than a thousand. Named entities (NEs) are crucial for many language technologies and NLP applications, including text comprehension, question answering, information retrieval and relation extraction. In this paper, we demonstrate the effectiveness of our MNE resource in two downstream tasks: knowledge graph augmentation and bilingual lexicon induction.

We extract our MNE resource from the Parallel Bible Corpus (PBC) (Mayer and Cysouw, 2014), a multiparallel corpus that covers more than 1300 languages. (Note however that we do not use Bible-specific features; therefore, our work is in principle applicable to any parallel corpus.) For some languages, PBC is the only available text (Wu et al., 2018). Multiparallel corpora contain sentence-level parallel text in more than two languages. Apart from PBC, JW300 (Agić and Vulić, 2019) and Tatoeba[1] are two other examples of such corpora. While the amount of data per language provided by highly multiparallel corpora is usually small, this type of corpus plays an important part in compiling resources for low-resource languages.

Creating a named entity resource is comparatively easy if sufficiently high-quality resources are available for a language. Such resources include named entity recognizers (Yadav and Bethard, 2018; Li et al., 2020); large monolingual corpora, which can be used to learn high-quality word embeddings or high-quality contextualized embeddings; parallel corpora that consist of large corpora (millions of words) per language (Lample et al., 2016; Ma and Hovy, 2016; Dasigi and Diab, 2011); or high-quality annotated data, e.g., training sets for named entity recognition (Wang and Manning, 2014; Wu et al., 2021; Wu et al., 2020) or implicit high-quality annotations like hyperlinks in Wikipedia (Tsai et al., 2016). Recent work (Wu et al., 2021; Li et al., 2021) with multilingual pretrained language models (PLMs) like BERT and XML-R for named entity recognition is promising, but also relies on moderately large monolingual corpora (e.g., a Wikipedia of decent size) to learn good quality contextualized representations. However, these monolingual corpora exist only for about 100 or so languages. For instance, Zulu is not included but we cover it in our experiments.

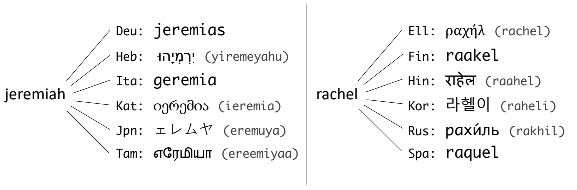


Figure 1: Two NEs from our resource, each with a sample of translations in six different languages

In this work, our goal is to cover the large number of languages for which these resources do not exist: no named entity recognizers, no large monolingual (or parallel) corpora, no annotated data (not even implicitly

[*] Now at Apple.

[1] https://tatoeba.org

annotated) and no pretrained language models (due to the lack of large monolingual corpora).

Many low-resource languages are covered in the PBC which gives us a chance to create resources for languages that currently do not have any – perhaps apart from an entry in the World Atlas of Language Structures (Dryer and Haspelmath, 2013) that is too abstract for most purposes in computational linguistics.

Since PBC is a parallel corpus, the question of why we do not use word alignment naturally arises. However, our experiments with word alignment on PBC were not successful for named entities. The reason is that word alignment performance deteriorates when parallel text is scarce (Och and Ney, 2003), especially for named entities as most are rare words. Our approach therefore does not depend on a word aligner and works well even when only a small parallel corpus is available. We directly compare with prior work that relies on word alignment.

Based on this motivation, we introduce CLC-BN (*Character Level Correspondence Bootstrapping and Neural transliteration*), a method for extracting a multilingual named entity resource from a parallel corpus, including in low-resource settings in which the available text per language in the corpus is small. CLC-BN learns a neural transliteration model from parallel-corpus statistics, without requiring any other bilingual resources, word aligners or seed data. In the first step, the method identifies NE correspondences in the parallel text. It then learn a neural transliteration model from these (noisy) NE correspondences. Finally, we use the learned model to identify high-confidence NE pairs in the parallel text. The first step (identifying NE correspondences) works at the character-ngram level, hence it is applicable to languages for which a tokenizer is not available, as opposed to word alignment based approaches. We will show that our method performs well for untokenized Japanese text.

In summary, our contributions are:

1. We present CLC-BN, a method that first identifies named entity correspondences in a parallel corpus and then learns a neural transliteration model from them.

2. We annotate a set of NEs to evaluate CLC-BN's performance on 13 languages through crowdsourcing and show a clear performance increase in comparison to prior work. We release the gold annotated sets as a resource for future work.[2]

3. Using CLC-BN, we create and release a named entity resource containing 674,493 names across 1340 languages, 503 names per language on average.[2]

4. For many languages, ours is the first published resource. We believe that it can be useful for future work in computational linguistics on the more than 1000 languages covered. We show experimentally that this is the case for knowledge graph augmentation and bilingual lexicon induction.

2. Related work

2.1. Word alignment

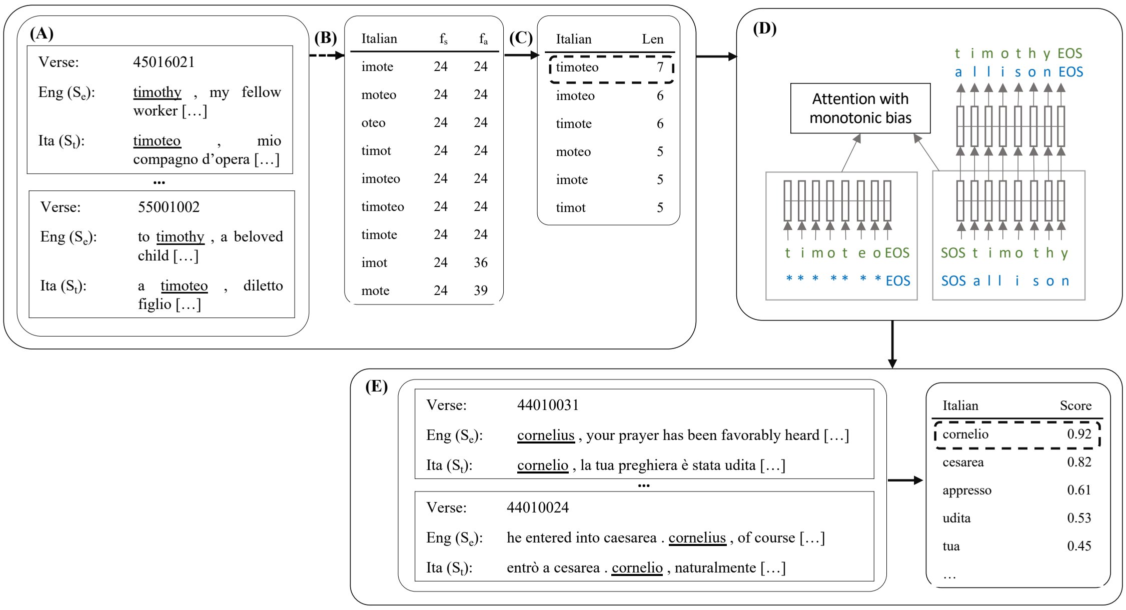
A multilingual named entity resource can be extracted from a parallel corpus via word alignment. Word alignment has been widely studied. Statistical word alignment models were introduced by Brown et al. (1993). More recently Giza++ (Och and Ney, 2000) and Eflomal (Östling et al., 2016) were released followed by neural network extensions (Ngo-Ho and Yvon, 2019). Other approaches use learned representations for creating alignments (Jalili Sabet et al., 2020). In concurrent work, Imani et al. (2021) have shown that better word alignment results can be achieved by exploiting multiparallel corpora. Previous work on named entity alignment and recognition uses combinations of alignment tools and postprocessing techniques. Dasigi and Diab (2011) use Giza++ for alignment and applied statistical machine translation (Koehn et al., 2007) and language-specific rules for improving transliteration. (Wu et al., 2018) use the Berkeley aligner (Liang et al., 2006) to word-align language pairs in the English Bible and further improve them with machine translation. In this paper, we do not use word aligners because of their low quality for named entities in small parallel corpora. We will directly compare with the word-alignment-based method of (Wu et al., 2018).

Recent approaches rely on parallel corpora and multilingual pre-trained models. Wu et al. (2021) construct a pseudo training set by performing translation and use multilingual BERT (Devlin et al., 2019a) to generate language independent features for training NER models. Li et al. (2021) use XLM-R (Conneau and Lample, 2019) to build an entity alignment model that projects English named entities into the parallel target language. While these approaches are promising, they are limited to the language set the models have been trained on (≈100). In contrast we apply CLC-BN to the more than one thousand languages in the Parallel Bible Corpus.

2.2. Transliteration

Prabhakar and Pal (2018) provide a comprehensive survey on transliteration. Recently, the task has been addressed with sequence-to-sequence models and transformers. Wu and Yarowsky (2018) perform experiments with these models on their Bible-based translation matrix dataset (Wu et al., 2018) and show that the task is challenging in the low-resource scenario. One of the causes is overfitting of the training set due to its reduced size. Our CLC-BN method uses a transliteration model and addresses this problem by augmenting the training set with monolingual target data (English) and introducing a monotonic bias.

[2] http://cistern.cis.lmu.de/ne_bible/



(A)
Verse: 45016021
Eng (S_e): timothy , my fellow worker […]
Ita (S_t): timoteo , mio compagno d'opera […]
...
Verse: 55001002
Eng (S_e): to timothy , a beloved child […]
Ita (S_t): a timoteo , diletto figlio […]
(B)
(C)
(D)
Attention with monotonic bias
t i m o t e o EOS
* * * * * * * EOS
SOS t i m o t h y
SOS a l l i s o n
t i m o t h y EOS
a l l i s o n EOS
(E)
Verse: 44010031
Eng (S_e): cornelius , your prayer has been favorably heard […]
Ita (S_t): cornelio , la tua preghiera è stata udita […]
...
Verse: 44010024
Eng (S_e): he entered into caesarea . cornelius , of course […]
Ita (S_t): entrò a cesarea . cornelio , naturalmente […]

Italian	f_s	f_a
imote	24	24
moteo	24	24
oteo	24	24
timot	24	24
imoteo	24	24
timoteo	24	24
timote	24	24
imot	24	36
mote	24	39

Italian	Len
timoteo	7
imoteo	6
timote	6
moteo	5
imote	5
timot	5

Italian	Score
cornelio	0.92
cesarea	0.82
appresso	0.61
udita	0.53
tua	0.45
…	

Figure 2: Data flow in CLC-BN. Example showing extraction of Italian NE training candidates for English "timothy" and identification of an Italian NE that matches English "cornelius". The input is the parallel corpus (A). CLC-B extracts from the parallel corpus ngrams that are candidate transliterations for "timothy" (B). These candidates are then filtered (C). (D): The architecture of the neural transliteration model. Green input-output pairs: Italian-English training data taken from the output of CLC-B. Blue input-output pairs: monolingual English training data. (E): We use the trained neural model to score candidates taken from the Italian parallel verses in which "cornelius" appears and keep the best scoring word.

2.3. Named entity resources

(Benites et al., 2020) introduce Translit, a transliteration resource created by combining and unifying public corpora. However, this dataset only covers 180 languages. BabelNet (Navigli and Ponzetto, 2012) is a multilingual encyclopedic dictionary that integrates WordNet, Wikipedia, GeoNames, inter alia. BabelNet is more comprehensive than other resources, but its NE coverage is still poor for many languages (e.g., for Inuktitut). We show in this paper that we can extend the coverage of BabelNet with our method. The Translation Matrix of (Wu et al., 2018) covers 591 languages. Their approach is based on word alignment. We show that our approach yields higher quality.

2.3.1. Named Entity Recognition resources

Named Entity Recognition (NER) systems usually require annotated data to achieve high accuracy. Our NE resource can be exploited to bootstrap such NER models for many different languages. (Al-Rfou et al., 2015) automatically extract named entities from Wikipedia link structure and Freebase attributes and create Polyglot-NER for 40 languages. (Pan et al., 2017) introduce WikiAnn, a resource for 282 Wikipedia languages that supports name tagging and entity linking. Our resource covers more than 1300 languages and CLC-BN does not rely on external sources other than the PBC.

2.4. Annotation projection

(Ehrmann et al., 2011) project annotations from English to five languages using a phrase-based statistical machine translation system and different methods: string matching, consonant signature matching and edit distance similarity. Ni et al. (2017) propose two methods for NER projection using heuristics, alignment information, and mapped word embeddings. Wang et al. (2018) describe a method for cross-lingual knowledge graph alignment of pre-aligned entities based on their distance in the learned embedding space. We project English NEs to the target languages exploiting character-level correspondence and a neural transliteration model without requiring any word alignment information or seed data.

2.5. Monotonicity

The performance of sequence-to-sequence models on some tasks can be improved by imposing an inductive bias of monotonicity (i.e., no character can be aligned to one that precedes a previously aligned character). Previous studies implement and analyze the effect of such a monotonic bias. Wu and Cotterell (2019) show that enforcing strict monotonicity and learning a latent alignment jointly while learning to transduce leads to improved performance for morphological inflection, transliteration, and grapheme-to-phoneme conversion. Rios et al. (2021) develop a general method for incorporating monotonicity into attention for seq2seq and Transformer models, agnostic of the task and model architectures. Similar to this prior work, we impose a monotonic bias on our neural transliteration model.

3. Method

We now describe CLC-BN.[3] Figure 2 shows architecture and data flow. For ease of development and evaluation, we also use the Uroman romanizer (Hermjakob et al., 2018). It converts scripts into Latin characters. But CLC-BN can be applied equally well without romanization. CLC-BN consists of two steps. First we extract character-level correspondences (CLC-B). Then we train a neural transliteration model to obtain the final set of named entities.

3.1. Character-Level Correspondence Bootstrapping (CLC-B)

We use cooccurrence statistics at the character level between English NEs and target language NEs to create a training set for the neural transliteration model. We use (Wu et al., 2018)'s list of English Bible NEs. NEs with frequency 1 are not considered in CLC-B because the FILTER step (#3 below) is likely to produce false positives (accidentally correlated ngrams) for them; but they are considered in §3.2.

CLC-B is designed based on the following simple correspondence assumption: if an English NE occurs in a verse, the corresponding target NE occurs in the parallel target verse and vice versa. This also implies that if N and M are the the number of verses in which the NE and its translation occur, then $N \approx M$. We do not require $N = M$ because we relax the correspondence assumption due to errors in the parallel corpus and due to the use of pronouns (including null pronouns, i.e., the pronoun is only present implicitly), which differs across languages.

We now describe our Character-Level Correspondence Bootstrapping (CLC-B) method, for the example of an English NE w. Algorithm 1 shows the pseudocode. Let f_a be the total frequency of an ngram in the target language and f_s its frequency in the subset of verses that contains w in English.

1. **EXTRACT.** (Line 4) Extract the parallel subcorpus that contains w from the parallel corpus. It consists of the English part S_e and the target language part S_t.

2. **GET_NGRAMS.** (Lines 5–13) For all character n-grams[4] ($3 < n < 20$) in S_t, determine f_s, the number of occurrences in S_t. Discard ngrams with $f_a > 50$ – this removes a small number of frequent NEs like Jesus, but avoids false positive matches with frequent ngrams. The resulting set of target ngrams is G_t.

3. **FILTER.** (Line 14) Filter G_t as follows. (a) Determine the ngram(s) with the highest f_s. Remove all other ngrams. (b) Determine the ngram(s) with the minimum absolute difference between f_a and f_s. Remove all other ngrams. Intuitively, most NEs in a particular domain are unique – so they should contain ngrams that only occur in the NE and not in other words. (c) Return the ngrams with the smallest length difference to w. This eliminates candidates that are much longer or shorter than w.

Algorithm 1 Pseudocode for the CLC-B method. Given a parallel corpus of English (E) and a target language (T), we identify, for each English NE, its target match. See §3.1 for details and for the EXTRACT and FILTER methods.

```
 1: procedure CLC-B(corpus $E$, corpus $T$, list English_NEs)
 2:   $pairs \leftarrow list()$
 3:   for w $\in$ English_NEs do
 4:     $S_e, S_t \leftarrow$ extract(w, $S, T$)                ▷ (1) EXTRACT
 5:     $G_t \leftarrow list()$
 6:     ngram_list $\leftarrow$ get_ngram_list($S_t$)
 7:     frequency_list $\leftarrow$ get_frequent_ngrams($S_t$)
 8:     for [ngram, count] $\in$ ngram_list do
 9:       if ngram $\in$ frequency_list or count $== 1$ then
10:         continue
11:       end if
12:       $G_t.append$([ngram, count])
13:     end for
14:     pairs.$append$(filter($G_t$))                        ▷ (3) FILTER
15:   end for
16:   return pairs
17: end procedure
```

3.2. Neural transliteration

CLC-B returns a noisy set of NE pairs, especially when only a small number of parallel verses is available for a language (we refer to this as the *lowest-resource* setting below). We build a neural sequence-to-sequence model (Sutskever et al., 2014) to refine it and to mine additional pairs. We use a single-layer bidirectional Gated Recurrent Unit (GRU) (Cho et al., 2014) encoder and a single-layer GRU decoder with attention (Luong et al., 2015). The sequences are processed at the character-level, with separate input and output vocabularies. Target language NEs are the input, English NEs the output; we use input/output when referring to the neural model (not source/target) because "target" in this paper refers to the target language that English is paired with.

To make best use of the limited training data in our experimental setup, we use augmentation and impose a monotonicity bias as described below. To avoid overfitting, we augment the training set with English NEs. We label the English Wikipedia dump[5] with the Flair Part-of-Speech tagger (Akbik et al., 2019), and select all NEs. We add, for each English NE mined from Wikipedia, one pair of the form (empty input NE, English output NE) to the training set. We use empty input NEs to prevent the learning of the identity function while helping the decoder to learn the structure of English words. To

[3] Reproducibility details in §A.

[4] We discard ngrams containing digits, punctuation and spaces.

[5] `https://dumps.wikimedia.org/` (01.04.2020)

	Lang	ISO	# verses	# parallel
low-resource languages	Arabic	Arb	31173	31062
	Finnish	Fin	31167	31061
	Greek	Ell	31183	31062
	Russian	Rus	31173	31062
	Spanish	Spa	31167	31062
	Swedish	Swe	31167	31062
	Zulu	Zul	31167	31062
lowest-resource languages	Hebrew	Heb	7952	7917
	Hindi	Hin	7952	7917
	Kannada	Kan	7952	7917
	Korean	Kor	7913	7869
	Georgian	Kat	4904	4844
	Tamil	Tam	7942	7917

Table 1: Number of verses in PBC and number of verses that are parallel with our English edition for the languages in our experiments. The English edition has 31,133 verses.

prevent generation of output independent of the input, we ensure equal proportions of original and augmented data by oversampling the former. Because transliterations are (with few exceptions) monotonic, we impose a monotonicity bias: we mask the attention matrix, so that the model cannot see anything to the left of the position previously attended to.

Given an English NE w and the verses S_e in which it appears, target candidates are all words in S_t, the verses parallel to S_e. Once the model is trained, we choose the best scoring candidate as w's transliteration where the score is the average log likelihood of the output characters (Severini et al., 2020).

We use a slightly different scoring step for non-tokenized languages (e.g., Japanese) because separated words in S_t are not available: given an English NE w, the target candidates are all ngrams that CLC-B has extracted for w in step 3b.

4. Evaluation and Analysis

We apply CLC-BN to the Parallel Bible Corpus (PBC) (Mayer and Cysouw, 2014) for evaluation and for creating our NE resource.[6] We evaluate on a subset of 13 languages that includes different scripts, resource availabilities and language families: Arabic, Greek, Finnish, Hebrew, Hindi, Kannada, Korean, Georgian, Russian, Spanish, Swedish, Tamil, and Zulu. These languages are also covered by the baselines and are therefore suitable for comparison. We view them as a representative subset for evaluating our method's performance. Note, however, that our NE resource covers all 1340 PBC languages: our approach is applicable to all languages since it does not use language-specific features and preprocessing steps.

PBC contains 1340 languages, most of which are low-resource. It is divided into subfiles, each containing Bible text from one language. Some languages that cover the Hebrew Bible and the New Testament completely contain about 30,000 verses. Other languages contain fewer than 8000 verses. We divide the languages into two categories: **lowest-resource**, fewer than 8000 verses; and **low-resource**, between 8000 and 32,000 verses.[7] Table 1 gives the number of verses for the editions we use. We evaluate our resource on human annotated data and on silver data with respect to the baselines and provide analysis.

4.1. Human evaluation

We annotated 60 NEs per language using Toloka,[8] a crowd-sourcing platform. Annotators had to pass an English test and successfully complete a training task to gain access to the annotation pool. Their performance was constantly checked using covert control questions. Each question contained the English NE and up to five possible options: one for each of the three baselines, one for CLC-B and one for CLC-BN. Each option consists of the word in the target script together with its romanized version in parentheses. Annotators had to mark all correct options that can be paired to the English NE, or none if no option is correct. Each question was annotated by exactly three annotators.

We calculate annotator agreement using Cohen's Kappa (Cohen, 1960), which measures agreement above chance. Similar to the setup of (Wu et al., 2018), we do not require that the annotators know the target languages. However, their average pairwise agreement is 0.73, "substantial agreement" according to Cohen's Kappa (Landis and Koch, 1977), indicating that they can find the correct corresponding target named entity even if they do not know the target language. To create the final gold set, we adopt a majority voting strategy and keep named entities that at least two annotators agreed on, resulting in at least 58 named entities per language. We evaluate CLC-BN and the baselines on this gold set.[9] The results can be found in Table 2, column "Hum". CLC-BN outperforms the baseline (Wu et al., 2018) for all languages (average difference of 7.9), with substantial improvements for the lowest-resource languages (difference of 21.1). The biggest improvements are for Hindi and Kannada (more than 30).

4.2. Silver evaluation

The gold dataset is used as the main evaluation of the resource. However, we additionally create a silver dataset to evaluate based on a larger set of hundreds of NEs. We create the silver set by translating each English NE to all target languages supported by the Google translation API[10] and comparing them with the NEs extracted

[6] Reproducibility details in §A

[7] "low-resource" is to be interpreted as referring to the setting in our experiments. For example, many resources are available for Russian, but in our setting we only use the Russian text that is available in PBC to evaluate how well our method works in a low-resource setting.

[8] https://toloka.yandex.com/

[9] We release the gold dataset to facilitate future research.

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

	Arb		Ell		Fin		Spa		Swe		Rus		Zul		AVG	
	Dist	Hum	Dist	Hum	Dist	Hum	Dist	Hum	Dist	Hum	Dist	Hum	Dist	Hum	Dist	Hum
(Wu et al., 2018)	67.9	70.0	47.2	80.0	89.0	90.0	87,6	91.7	88.8	88.3	60.6	72.9	61.9	84.8	65.9	82.5
Östling et al. (2016)	69.8	61.7	53.4	88.3	77.7	76.7	83.9	86.7	81.2	85.0	64.8	83.1	52.9	86.4	60.9	81.1
Sabet et al. (2020)	18.1	20.0	23.5	40.0	49.8	60.0	35.6	45.0	41.6	50.0	39.6	45.8	18.3	25.4	29.6	40.9
CLC-B	53.8	56.7	32.2	45.0	59.9	50.0	48.0	48.3	52.0	48.3	46.5	57.6	55.1	74.6	46.6	54.4
CLC-BN	70.6	81.7	54.7	91.7	86.5	93.3	89.6	96.7	89.9	91.7	70.2	84.8	68.8	93.2	71.9	90,4

	Heb		Hin		Kan		Kat		Kor		Tam		AVG	
	Dist	Hum	Dist	Hum	Dist	Hum	Dist	Hum	Dist	Hum	Dist	Hum	Dist	Hum
(Wu et al., 2018)	53.4	62.5	64.1∗	76.3∗	41.5	61.7	64.3	70.0	30.5	54.2	47.1∗	66.1∗	50.2	65.1
Östling et al. (2016)	65.5	83.9	57.7∗	69.5∗	23.1	38.3	64.0	68.3	16.6	33.9	20.5∗	35.6∗	41.2	57.5
Sabet et al. (2020)	27.8	23.2	41.6∗	47.5∗	26.5	46.7	28.0	20.0	23.5	40.0	30.4∗	47.5∗	29.6	37.5
CLC-B	37.2	51.8	43.1∗	39.0∗	30.7	48.3	41.7	45.0	23.8	37.3	38.6∗	47.5∗	35.9	44.8
CLC-BN	62.6	71.4	78.6∗	94.9∗	45.9	93.3	70.8	88.3	34.2	78.0	59.5∗	91.5∗	58.6	86.2

Table 2: Precision of NE correspondence identification for low-resource (top: Hebrew Bible and New Testament) and lowest-resource (bottom: New Testament only) languages. We compare Translation Matrix (Wu et al., 2018), Eflomal (Östling et al., 2016), SimAlign (Sabet et al., 2020), CLC-B and CLC-BN. Comparisons are with silver data+Jaro distance (Dist) and with gold human annotated data (Hum). *: evaluation on romanization for fair comparison with baselines.

by CLC-BN using the Jaro distance (Jaro, 1989). The distance takes into account the number and order of characters shared by two strings; e.g., the NE "salome" has a distance of 0.05 from "salom" and 0.11 from "calom". Jaro is frequently used for entity matching and is well-suited for short strings (Cohen et al., 2003). We use a threshold of 0.3 for the Jaro distance, chosen to be strict enough to evaluate the NEs and to take into account noise in the pairs produced by Google Translate. For example, the silver translation of "jannes" in Greek is γιάννες (giánnes) while our data contains ιαννής (iannís), which is also correct; their distance is 0.26. Another example is the name "mitylene" that the silver data translates to μυτυλένιο (mytylénio) and has a distance of 0.27 to our translation μυτιλήνη (mytilíni). By design, the silver data provides only a single translation for each English NE. However, multiple translations are often correct, due to the variability of morphology, transliteration, naming conventions and dialects (Prabhakar and Pal, 2018). For example, the English NE "Paul" can be aligned to Russian "Pavel" and "Pavla". For this reason, our results on the silver standard must be interpreted as lower bounds.

Arabic and Hebrew are standardly written without short vowels. This is also the case for the silver data. However, some PBC editions are written with short vowels, so we postprocess predictions by removing short vowel diacritics.

Table 2 shows results for the 13 languages. The ranking of baselines and methods is similar to the one obtained with the gold human evaluation with CLC-BN being always the best, except for Finnish. Improvements for lowest-resource languages (lower part of the table) are large, up to 48% difference on average. CLC-BN outperforms (Wu et al., 2018) for 12 of the 13 languages.[11]

4.3. Word alignment comparison

NE correspondences can also be obtained using a word aligner. We compare our results with pairs obtained using Eflomal (Östling et al., 2016), a statistical word aligner, and SimAlign (Sabet et al., 2020), a high-quality word aligner that leverages multilingual word embeddings. Table 2 shows precision for silver and gold data. CLC-BN outperforms Eflomal (with the exception of Hebrew) and SimAlign for all 13 languages. We attribute this to the fact that NEs are hard to word-align because most of them are infrequent, resulting in alignment errors due to sparseness. CLC-BN could be integrated into word alignment pipelines to boost word aligner performance for NEs (Sajjad et al., 2011; Semmar and Saadane, 2013).

CLC-B works at the character level and is applicable to non-tokenized languages while aligners are not. Japanese is non-tokenized, so we evaluate it (only for CLC-B and CLC-BN since the other methods were not run on Japanese). We evaluate the 979 pairs of CLC-BN with the silver data and obtain a precision of 63.2%. We also use Toloka for the gold evaluation of 60 random pairs and obtain a precision of 60%. However, in this case each question has at most two options (CLC-B and CLC-BN – in contrast to five as for 4.1), which can hinder the annotators' judgments having less comparison terms. For this reason, we also asked three experts to evaluate the 60 pairs and obtained a precision of 85%.

4.4. Impact of corpus size

Table 2 shows that precision for lowest-resource languages (less than 8000 verses, bottom) is worse than those for low-resource languages (about 30,000 verses, top), with an average difference of 13.3% for silver data, and 4.2% for gold data. The small gap on gold data, highlights that our method is appropriate also for the lowest-resource setting. Table 3 shows some examples of aligned pairs according to CLC-BN. We see that errors arise as the frequency of NEs in the English corpus diminishes. For example, the Kannada alignment for

[11] The exception is Finnish, which is probably due to the fact that machine translation (which was used for (Wu et al., 2018)) performs well for high-resource languages. Note however, that CLC-BN performs best for Finnish in the (more reliable) human ("Hum") evaluation.

#	English	Arabic	Finnish	Greek	Hebrew	Kannada	Russian	Tamil
28	elijah	alalihaau	eliaa	elia	veaeliyahu	eliiyanaagali	elisei	eliyaavaa
12	titus	tiytusa	titus	titos	titos	titanannu	titu	tiittuvin
8	elizabeth	aaliysaabaata	elisabet	elisabet	elisheva	elisabeet	elizaveta	elicapet
3	miletus	miyliytusa	miletokseen	mileto	lemilitos	mileetakke	mileta	mileettu
2	rufus	ruwfusa	rufuksen	roufo	vishelom	uphaniguu	rufa	ruupuvukkum
2	hermes	wahirmisa	hermeeksi	epairne	heremes	meeyaniguu	germes	ermee

Table 3: Examples of named entity alignments (romanized). "#" column shows the number of verses in which the English word appears.

Lang	Eng	Freq	CLC-B	CLC-BN
Arb	anah	10	الشيخة (alshaykha)	أنا (ana)
Rus	joanna	2	мария (mariya)	иоа́нна (joanna)
Fin	perez	2	hesroni	peresin
Kan	cainan	2	ನಾನಾ (naanaa)	ಕಯಿನಾನನ (kayinaanana)
Tam	azor	2	எலியூதுக்குத் (eliyuutukkut)	ஆசோர் (aacoor)

Table 4: Examples of improvement due to neural transliteration. CLC-B: incorrect prediction of CLC-B. CLC-BN: correct prediction obtained with neural transliteration.

"rufus" and Greek and Kannada alignments for "hermes" are incorrect. Both words are short, indicating another source of errors: short words provide less of a signal for the neural transliteration model than long words do.

4.5. Impact of neural transliteration

Table 2 shows precision for CLC-B and CLC-BN. All languages benefit from neural transliteration with an average improvement of 30.9 percentage points. One of the reasons is that CLC-B was designed to discard English NEs that appear only once in the corpus. Table 4 shows examples where neural transliteration corrects an error made by CLC-B. Most of these cases have low frequency. This is not surprising as the risk of false positives increases as the frequency decreases because the heuristics used in CLC-B (§3.1) are less reliable for low-frequency NEs.

4.6. Error analysis

In our manual error analysis, we found two main types of errors.

(1) The neural model generally learns well how to transliterate the beginning of a word, but error rates are higher word-internally. For example, the NE "balak" is wrongly paired to "pileeyaam" instead of "paalaak" and "menna" is paired to "meleyaa" instead of "meyinaan" in Tamil. The neural model has to learn two aspects of transliteration: transliteration proper (i.e., character correspondences) and alignment. This type of error indicates that alignment performance should be improved. In future work, we plan to explore neural architectures that more explicitly model the problem as alignment.

(2) For some low-resource languages, the output of CLC-B has a high level of noise, so the neural model fails to learn some character correspondences. In some cases, the output of the neural model is unrelated to the input. This type of error indicates that the CLC-B method should be improved further. As shown in Tables 3 and 4, low-frequency words contain more errors. In future work, we plan to adopt an iterative strategy that considers gradually more and more named entities, starting with the most confident ones.

5. Use cases

5.1. Transliteration

A straightforward application of our named entity resource, as described by (Wu et al., 2018), is to create transliteration models. They showed that a character-based Moses SMT system trained over a dataset of named entities extracted from the Bible (whose performance is lower than our method's, based on Table 2) performs better than a Unicode baseline. We now present two additional applications of our named entity resource: extending existing multilingual dictionaries and cross-lingual mapping of word embeddings.

5.2. Extending existing multilingual resources

BabelNet[12] (Navigli and Ponzetto, 2012) is a multilingual encyclopedic dictionary. It was created by integrating more than 35 WordNets, covering 500 languages, and has about 20 million entries.

We want to show that one can use our resource to enrich BabelNet further. Since CLC-BN covers many more languages than BabelNet, we can simply extend BabelNet by adding more languages like Burarra, North Junín Quechua, and Mian to it. Regarding the languages that BabelNet already supports, we check whether we can add more entries exploiting our resource. To this end, for each word pair (English:target-language) in CLC-BN, we check whether a translation of the English word exists in BabelNet in the target language. Results are depicted in Table 5. On average, 27% (i.e., 206 words) of the English words have no correspondence in the target language. These are mostly rare words that are difficult to translate without accessing a resource as rich as PBC. From a manual investigation, we find that our resource could also help to improve the quality of BabelNet; some translations of the latter are completely incorrect or wrongly written with Latin characters. Examples for Greek are hamor/εμμώρ (emmor), which

[12] `https://babelnet.org/`

Lang.	CLC-BN	Babel	New NEs	New NEs %
Arb	977	683	294	30.1
Fin	979	647	332	33.9
Ell	979	658	321	32.8
Rus	485	449	36	7.4
Spa	979	784	195	19.9
Swe	979	684	295	30.1
Zul	979	471	508	51.9
Heb	467	413	54	11.6
Hin	467	334	133	28.5
Kan	467	299	168	36.0
Kor	467	386	81	17.3
Kat	368	271	97	26.4
Tam	433	318	115	26.6
Jpn	979	715	264	27.0
Zho	979	698	281	28.7
Tha	467	337	130	27.8
AVG.	715	509	206	27.2

Table 5: Extension of BabelNet with named entities based on our resource. Example (first line, "Arb"). CLC-BN returns 977 English-Arabic NE pairs. BabelNet contains Arabic translations for 683 of these English NEs, but 294 (30.1%) lack an Arabic translation. Thus we add 294 English-Arabic NE pairs that were not covered by BabelNet.

BabelNet translates as Δείνα (Deina), and ethan/εϑάν, incorrectly transliterated with Latin characters.

5.3. Cross-lingual mapping of word embeddings

An effective method for creating bilingual word embeddings is to train word embeddings for each language independently using monolingual resources and then aligning them using a linear transformation (Artetxe et al., 2018). Approaches for word embedding alignment can be grouped into three categories: supervised (Mikolov et al., 2013; Lazaridou et al., 2015), semisupervised (Artetxe et al., 2017) and unsupervised (Artetxe et al., 2018; Alvarez-Melis and Jaakkola, 2018). Supervised approaches require a bilingual dictionary with a few thousand entries to learn the mapping. Semisupervised procedures need a small seed dictionary. Unsupervised approaches can align word embeddings without any bilingual data but, as shown by Vulić et al. (2019), they are only effective when the two languages are similar enough, restricting their applicability.

In this use case, we use our resource as the initial seed dictionary for semisupervised alignment of word embeddings for language pairs where unsupervised methods fail. We select three such language pairs – English/Japanese, English/Chinese and English/Tamil – and show that VecMap,[13] a semisupervised method, can successfully employ our NE resource to align these languages. VecMap implements the method proposed by Artetxe et al. (2018), which is a state-of-the-art method for unsupervised cross-lingual word embedding mapping. It creates an initial set of word pairings based on the distribution of words in their similarity matrix. Then it employs a self-learning method to improve the mapping iteratively.

	Eng-Jpn	Eng-Tam	Eng-Zho
Unsupervised	0.0	0.0	0.0
Semisupervised	30.43	14.4	30.1

Table 6: P@1 BLI results with unsupervised VecMap compared to semisupervised VecMap, which uses our NE resource for initialization

We evaluate the embeddings on the Bilingual Lexicon Induction (BLI) task and the gold dataset provided by MUSE (Conneau et al., 2018). We use Wikipedia fastText embeddings (Bojanowski et al., 2017) as monolingual input vectors and report precision at one (P@1) for the unsupervised and semisupervised approaches in Table 6. While the fully unsupervised method fails to align these languages, the semisupervised approach based on our resource has much better results confirming that our NE resource can be effectively used as seed data.

6. Resource

We release a resource of named entities for 1340 languages, 1134 of which are lowest-resource.[14] The resource mainly contains people and location NEs. The total number of NEs is 674,493, so there are 503 NEs per language on average with at least 300 NEs in 95% of the languages. The three best represented language families (Dryer and Haspelmath, 2013) are Austronesian, Niger-Congo and Indo-European. However, our coverage broadly includes all major areas of linguistic diversity, including Amazonian (e.g., Kaingang), African (e.g., Sango) and Papua New Guinea (e.g., Saniyo-Hiyewe).

7. Conclusion

We presented CLC-BN, a new method that identifies named entity correspondences and trains a neural transliteration model on them. CLC-BN does not need any other bilingual resources beyond the parallel corpus nor a word aligner or seed data. We showed that it outperforms prior work on silver data and human-annotated gold data. We created a new NE resource for 1340 languages by applying CLC-BN to the Parallel Bible Corpus and illustrated its utility by demonstrating good performance on two downstream tasks: knowledge graph augmentation and bilingual lexicon induction.

Acknowledgments. This work was funded by the European Research Council (grant #740516) and the German Federal Ministry of Education and Research (BMBF, grant #01IS18036A).

8. Bibliographical References

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL*

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

[14] Our NEs resource is freely available at `http://cistern.cis.lmu.de/ne_bible/`

2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pages 54–59.

Alvarez-Melis, D. and Jaakkola, T. S. (2018). Gromov-wasserstein alignment of word embedding spaces. *arXiv preprint arXiv:1809.00013*.

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

Artetxe, M., Labaka, G., and Agirre, E. (2018). 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, Melbourne, Australia, July. Association for Computational Linguistics.

Bird, S. (2020). Decolonising speech and language technology. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3504–3519.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). 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)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Cohen, W., Ravikumar, P., and Fienberg, S. (2003). A comparison of string metrics for matching names and records. In *Kdd workshop on data cleaning and object consolidation*, volume 3, pages 73–78.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Dasigi, P. and Diab, M. (2011). Named entity transliteration generation leveraging statistical machine translation technology. In *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, pages 106–111.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019a). 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)*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019b). 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, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Imani, A., Sabet, M., Keremşenel, L., Dufter, P., Yvon, F., and Schütze, H. (2021). Graph algorithms for multiparallel word alignment. In *The 2021 Conference on Empirical Methods in Natural Language Processing*.

Jalili Sabet, M., Dufter, P., Yvon, F., and Schütze, H. (2020). SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online, November. Association for Computational Linguistics.

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420.

Joshi, P., Santy, S., Budhiraja, A., Bali, K., and Choudhury, M. (2020). The state and fate of linguistic diversity and inclusion in the nlp world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

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

Li, J., Sun, A., Han, J., and Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Li, B., He, Y., and Xu, W. (2021). Cross-lingual named entity recognition using parallel corpus: A new approach using xlm-roberta alignment. *arXiv preprint arXiv:2101.11112*.

Liang, P., Taskar, B., and Klein, D. (2006). Alignment by agreement. In *Proceedings of the Human Lan-*

guage Technology Conference of the NAACL, Main Conference, pages 104–111.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

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

Ngo-Ho, A.-K. and Yvon, F. (2019). Neural baselines for word alignments. In *International Workshop on Spoken Language Translation*.

Ni, J., Dinu, G., and Florian, R. (2017). Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1470–1480.

Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 440–447.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Östling, R., Tiedemann, J., et al. (2016). Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics*.

Prabhakar, D. K. and Pal, S. (2018). Machine transliteration and transliterated text retrieval: a survey. *Sādhanā*, 43(6):1–25.

Rios, A., Amrhein, C., Aepli, N., and Sennrich, R. (2021). On biasing transformer attention towards monotonicity. In *2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Sabet, M. J., Dufter, P., Yvon, F., and Schütze, H. (2020). Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643.

Sajjad, H., Fraser, A., and Schmid, H. (2011). An algorithm for unsupervised transliteration mining with an application to word alignment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 430–439.

Semmar, N. and Saadane, H. (2013). Using transliteration of proper names from Arabic to Latin script to improve English-Arabic word alignment. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1022–1026, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

Severini, S., Hangya, V., Fraser, A., and Schütze, H. (2020). Combining word embeddings with bilingual orthography embeddings for bilingual dictionary induction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6044–6055, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Tsai, C.-T., Mayhew, S., and Roth, D. (2016). Cross-lingual named entity recognition via wikification. In *Proceedings ofThe 20th SIGNLL Conference on Computational Natural Language Learning*, pages 219–228.

Vulić, I., Glavaš, G., Reichart, R., and Korhonen, A. (2019). Do we really need fully unsupervised cross-lingual embeddings? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4407–4418, Hong Kong, China, November. Association for Computational Linguistics.

Wang, M. and Manning, C. D. (2014). Cross-lingual projected expectation regularization for weakly supervised learning. *Transactions of the Association for Computational Linguistics*, 2:55–66.

Wang, Z., Lv, Q., Lan, X., and Zhang, Y. (2018). Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 349–357.

Wu, S. and Cotterell, R. (2019). Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537.

Wu, W. and Yarowsky, D. (2018). A comparative study of extremely low-resource transliteration of the world's languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Wu, Q., Lin, Z., Wang, G., Chen, H., Karlsson, B. F., Huang, B., and Lin, C.-Y. (2020). Enhanced meta-learning for cross-lingual named entity recognition with minimal resources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9274–9281.

Wu, Q., Lin, Z., Karlsson, B. F., Huang, B., and Lou, J.-G. (2021). Unitrans: unifying model transfer and data transfer for cross-lingual named entity recognition with unlabeled data. In *Proceedings of the*

Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pages 3926–3932.

Yadav, V. and Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158.

9. Language Resource References

Agić, Ž. and Vulić, I. (2019). Jw300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210.

Al-Rfou, R., Kulkarni, V., Perozzi, B., and Skiena, S. (2015). Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 586–594. SIAM.

Benites, Fernando and Duivesteijn, Gilbert François and von Däniken, Pius and Cieliebak, Mark. (2020). *TRANSLIT: A Large-scale Name Transliteration Resource*. European Language Resources Association.

Bojanowski, Piotr and Grave, Edouard and Joulin, Armand and Mikolov, Tomas. (2017). *Enriching Word Vectors with Subword Information*.

Alexis Conneau and Guillaume Lample and Marc'Aurelio Ranzato and Ludovic Denoyer and Herv'e J'egou. (2018). *Word Translation Without Parallel Data*.

Matthew S. Dryer and Martin Haspelmath. (2013). *WALS Online*. Max Planck Institute for Evolutionary Anthropology.

Ehrmann, M., Turchi, M., and Steinberger, R. (2011). Building a multilingual named entity-annotated corpus using annotation projection. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 118–124.

Hermjakob, Ulf and May, Jonathan and Knight, Kevin. (2018). *Out-of-the-box universal romanization tool uroman*.

Mayer, Thomas and Cysouw, Michael. (2014). *Creating a massively parallel bible corpus*.

Navigli, Roberto and Ponzetto, Simone Paolo. (2012). *BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network*. Elsevier.

Pan, Xiaoman and Zhang, Boliang and May, Jonathan and Nothman, Joel and Knight, Kevin and Ji, Heng. (2017). *Cross-lingual name tagging and linking for 282 languages*.

Wu, Winston and Vyas, Nidhi and Yarowsky, David. (2018). *Creating a translation matrix of the Bible's names across 591 languages*.

A. Reproducibility Information

We run our method on up to 48 cores of Intel(R) Xeon(R) CPU E7-8857 v2 with 1TB memory and a single GeForce GTX 1080 GPU with 8GB memory. CLC-BN is implemented in Python and takes approximately 2 minutes to run for one language. The neural model is implemented in PyTorch and has one encoder and one decoder layer (batch size 16, hidden layer size 32, learning rate 0.01, dropout 0.4, 24K parameters). We use Luong et al. (2015)'s attention. Each training of the neural transliteration model requires at most 10 minutes. SimAlign (Sabet et al., 2020) alignments are obtained using multilingual BERT (Devlin et al., 2019b). We use subword alignments and the forward alignment to ensure that all English NEs are aligned. Eflomal (Östling et al., 2016) alignments are obtained with default parameters and the forward alignment. The Jaro distance is calculated using the Python library textdistance.[15]
For the cross-lingual word alignment experiment we used the latest VecMap code available in its git repository[16] (no snapshot is available). We ran it using the $<--unsupervised>$ and $<--semi\_supervised>$ switches. All other parameters are left as their default value. The monolingual word alignments are downloaded from fastText's official website.[17]

[15] `https://pypi.org/project/textdistance/`

[16] commit ID: `b82246f6c249633039f67fa6156e51d852bd73a3`

[17] `https://fasttext.cc/docs/en/pretrained-vectors.html`

Chapter 6

Corresponds to the following publication:

> Ayyoob Imani*, **Silvia Severini***, Masoud Jalili Sabet, François Yvon, and Hinrich Schütze. “Graph-Based Multilingual Label Propagation for Low-Resource Part-of-Speech Tagging.”. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. *Equal contribution.

Resource: `https://github.com/ayyoobimani/GLP-POS`

Declaration of Co-Authorship: Ayyoob Imani conceived the initial research contribution. I contributed with multiple ideas on the GNN model such as the integration of type-level information and the selection of training languages. I conceived the neural POS tagger and all the evaluations in the paper. I regularly discussed the work with my advisor Hinrich Schütze. Ayyoob Imani and I wrote the initial draft. All authors helped review the final draft of the paper and gave advice.

Graph-Based Multilingual Label Propagation for Low-Resource Part-of-Speech Tagging

Ayyoob Imani [*1], **Silvia Severini**[*1],
Masoud Jalili Sabet[1], **François Yvon**[2], **Hinrich Schütze**[1]
[1]Center for Information and Language Processing (CIS), LMU Munich, Germany
[2]Université Paris-Saclay, CNRS, LISN, France
{ayyoob, silvia, masoud}@cis.lmu.de, francois.yvon@limsi.fr

Abstract

Part-of-Speech (POS) tagging is an important component of the NLP pipeline, but many low-resource languages lack labeled data for training. An established method for training a POS tagger in such a scenario is to create a labeled training set by transferring from high-resource languages. In this paper, we propose a novel method for transferring labels from multiple high-resource source to low-resource target languages. We formalize POS tag projection as graph-based label propagation. Given translations of a sentence in multiple languages, we create a graph with words as nodes and alignment links as edges by aligning words for all language pairs. We then propagate node labels from source to target using a Graph Neural Network augmented with transformer layers. We show that our propagation creates training sets that allow us to train POS taggers for a diverse set of languages. When combined with enhanced contextualized embeddings, our method achieves a new state-of-the-art for unsupervised POS tagging of low resource languages.

1 Introduction

In many applications, Part-of-Speech (POS) tagging is an important part of the NLP pipeline. In recent years, high-accuracy POS taggers have been developed owing to advances in machine learning methods that combine pretraining on large unlabeled corpora and supervised fine-tuning on well-curated annotated datasets. This methodology only applies to a handful of high-resource (HR) languages for which the necessary training data exists, leaving behind the majority of low-resource (LR) languages. When training resources are scarce, an established method for training POS taggers is to automatically generate the training data via cross-lingual transfer (Yarowsky and Ngai, 2001; Fossum and Abney, 2005; Agić et al., 2016; Eskander et al., 2020). Typically, POS annotations are projected through alignment links from the HR source to the LR target of a word aligned parallel corpus.

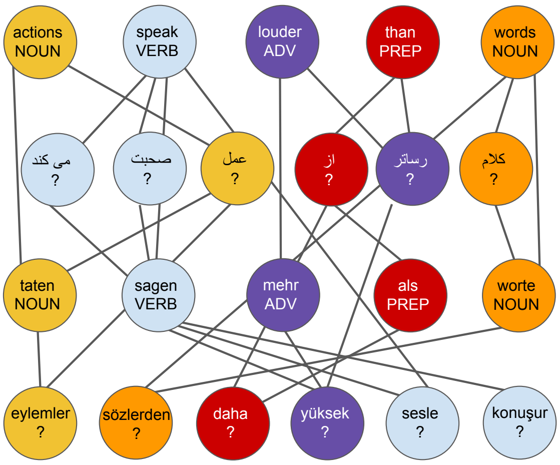


Figure 1: The sentence "actions speak louder than words" in English and its translations to Persian, German, and Turkish, aligned at the word level. The POS tags for high-resource English and German are known. We use a GNN to exploit this graph structure and compute POS tags for low-resource Persian and Turkish.

In this paper, we propose **GLP** (**G**raph **L**abel **P**ropagation), a novel method for transferring labels simultaneously *from multiple high-resource source languages to multiple low-resource target languages*. We formalize POS tag projection as graph-based label propagation. Given translations of a sentence in multiple languages, we create a graph with words as nodes and alignment links as edges by aligning words for all language pairs. We then propagate POS labels associated with source language nodes to target language nodes, using a label propagation model that is formalized as a Graph Neural Network (GNN) (Scarselli et al., 2008). Nodes are represented by a diverse set of features that describe both linguistic properties and graph structural information. In a second step, we additionally employ self-learning to obtain reliable

[*]Equal contribution.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1577 - 1589
December 7-11, 2022 ©2022 Association for Computational Linguistics

training instances in the target languages.

Our approach is based on *multiparallel corpora*, meaning that the translation of each sentence is available in more than two languages. We exploit the Parallel Bible Corpus (PBC) of Mayer and Cysouw (2014),[1] a multiparallel corpus covering more than 1000 languages, many of which are extremely low-resource, by which we mean that only a tiny amount of unlabeled data is available or that no language technologies exist for them at all (Joshi et al., 2020).

We evaluate our method on a diverse set of low-resource languages from multiple language families, including four languages not covered by pre-trained language models (PLMs). We train POS tagging models for these languages and evaluate them against references from the Universal Dependencies corpus (Zeman et al., 2019). We compare the results of our method against multiple state-of-the-art (SOTA) cross-lingual unsupervised and semisupervised POS taggers employing different approaches like annotation projection and zero-shot transfer. Our experiments highlight the benefits of our new transfer and self-learning methods; crucially, they show that reasonably accurate POS taggers can be bootstrapped without any annotated data for a diverse set of low-resource languages, establishing a new SOTA for high-resource-to-low-resource cross-lingual POS transfer. We also assess the quality of the projected annotations with respect to "silver" references and perform an ablation study. To summarize, our contributions are:[2]

- We formalize annotation projection as graph-based label propagation and introduce two new POS annotation projection models, GLP-B (GLP-Base) and GLP-SL (GLP-SelfLearning).

- We evaluate GLP-B and GLP-SL on 17 low-resource languages, including 4 languages not covered by large PLMs.

- By comparing our method with various supervised, semisupervised, and PLM-based approaches for POS tagging of low-resource languages, we establish a new SOTA for unsupervised POS tagging.

2 Related work

POS tagging Part of Speech tagging aims to assign each word the proper syntactic tag in context (Manning and Schütze, 1999). For high-resource languages, for which large labeled training sets are available, high-accuracy POS tagging is achieved through supervised learning (Kondratyuk and Straka, 2019; Tsai et al., 2019).

Zero-shot transfer In low-resource settings, one approach is to use cross-lingual transfer thanks to pretrained multilingual representations, thereby enabling zero-shot POS tagging. Kondratyuk and Straka (2019) analyze the few-shot and zero-shot performance of mBERT (Devlin et al., 2019) fine-tuning on POS tagging. We include this approach in our set of baselines below. Ebrahimi and Kann (2021) and Wang et al. (2022) analyze zero-shot POS tagging performance of XLM-RoBERTa (Conneau et al., 2020) and propose complementary methods such as continued pretraining, vocabulary expansion and adapter modules for better performance. We show that combining GLP with Wang et al. (2022)'s embeddings further improves our base performance.

Annotation projection Annotation projection is another approach to annotating low-resource languages. Yarowsky and Ngai (2001) first proposed projecting annotation labels across languages, exploiting parallel corpora and word alignment. To reduce systematic transfer errors, Fossum and Abney (2005) extended this by projecting from multiple source languages. Agić et al. (2015a) and Agić et al. (2016) exploit multilingual transfer setups to bootstrap POS taggers for low-resource languages starting from a parallel corpus and taggers and parsers for high-resource languages. Other works project labels by leveraging token and type-level constraints (Täckström et al., 2013; Buys and Botha, 2016a; Eskander et al., 2020). The latter study notably proposes an unsupervised method for selecting training instances via cross-lingual projection and trains POS taggers exploiting contextualized word embeddings, affix embeddings and hierarchical Brown clusters (Brown et al., 1992). This approach is also used as a baseline below.

Semi-supervised approaches have been proposed to mitigate the noise of projecting between languages. This can be achieved with auxiliary lexical resources (Täckström et al., 2013; Ganchev and Das, 2013; Wisniewski et al., 2014; Li et al.,

[1] We do not use PBC-specific features. Thus, our work is in principle applicable to any multiparallel corpus.

[2] Our code, data, and trained models are available at `https://github.com/ayyoobimani/GLP-POS`

2012) that guide unsupervised learning or act as an additional training signal (Plank and Agić, 2018). Other works combine manual and projected annotations (Garrette and Baldridge, 2013; Fang and Cohn, 2016). We outperform prior works without the use of additional resources such as dictionaries or annotations.

Graph Neural Networks Many natural and real-life structures like physical systems, social networks & interactions, and molecular fingerprints have a graph structure (Liu and Zhou, 2020). Graph neural networks have been successfully used to model them. Applications include social spammer detection (Wu et al., 2020), learning molecular fingerprints (Duvenaud et al., 2015) and human motion prediction (Li et al., 2020). Recently, GNNs have been adopted for NLP tasks such as text classification (Peng et al., 2018), sequence labeling (Zhang et al., 2018; Marcheggiani and Titov, 2017), neural machine translation (Bastings et al., 2017; Beck et al., 2018), and alignment link prediction (Imani et al., 2022). As far as we know, our work is the first to formalize the annotation projection problem as graph-based label propagation.

Multiparallel corpora A multiparallel corpus provides the translations of a source text in more than two languages. A few such corpora (Agić and Vulić, 2019; Mayer and Cysouw, 2014; Tiedemann, 2012) provide sentence-level aligned text for hundreds or thousands of languages; for many of these languages only a tiny amount of digitized content is available (Joshi et al., 2020). Although the amount of text found in existing multiparallel corpora is far less than in monolingual corpora, we believe that they can serve as cross-lingual bridges, with which effective representation for low-resource languages can be derived. Highly multiparallel corpora have been used for expanding pretrained models to more languages (Ebrahimi and Kann, 2021; Wang et al., 2022), word alignment improvement and visualization (ImaniGooghari et al., 2021; Imani et al., 2022), embedding learning (Dufter et al., 2018), and annotation projection (Agić et al., 2015b; Severini et al., 2022).

3 Method

We now introduce our *Graph Label Propagation* (GLP) method, which formalizes the problem of annotation projection as graph-based label propagation. We first describe the graph structure, then the features associated with each node, and finally the architecture of our model.

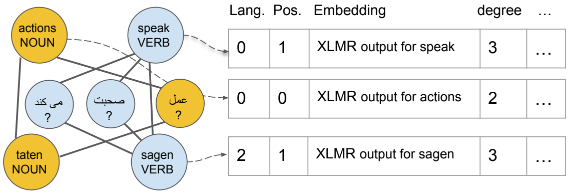


Figure 2: An example of how we represent nodes of an alignment graph using features for a part of the graph in Figure 1.

3.1 Problem formalization

The *multilingual alignment graph* (MAG) of a sentence is formalized as follows. Each sentence σ in our multiparallel corpus exists in a set L of languages.[3] L contains both high-resource source languages (in L_s) and low-resource target languages (in L_t) with $L_s \cup L_t = L$. Each word in these $|L|$ versions of σ will constitute a node in our graph. We first automatically annotate the text in all the source languages using pre-existing taggers: these POS tags are node labels; they are only known for languages in L_s, unknown otherwise. We then use Eflomal (Östling and Tiedemann, 2016), an unsupervised word alignment tool to compute alignment links for all $\frac{|L|*(|L|-1)}{2}$ language pairs: these links define the edges of our MAG. Figure 1 displays an example MAG for four languages, with English and German as sources and Turkish and Persian as targets. Note that both the word alignments and the node labels are noisy, since we do not use gold data but statistical methods to generate them.

3.2 Features

To train graph neural networks, we represent each node using a set of features (Duong et al., 2019). In Figure 2, you see a simple illustration of how nodes are represented using a feature vector. The graph in this figure is part of the original graph in Figure 1. Two types of features are considered: features that represent the inherent meaning of a node/word (word representation features) and features that describe the position of a node within the graph (graph structural features). Node representation features consist of: XLM-R (Conneau et al., 2020) embeddings, the node's language and its position within the sentence. Since XLM-R embeddings are not available for all languages, we alternatively

[3] $|L|$ might be different for different sentences.

experiment with static word embeddings created using Levy et al. (2017)'s sentence-ID method, which we train on PBC. Our graph structural features are similar to Imani et al. (2022)'s work on link prediction. They include five centrality features: *degree, closeness* (Freeman, 1978), *betweenness* (Brandes, 2001), *load* (Newman, 2001), and *harmonic centrality* (Boldi and Vigna, 2014). Each of these features describes the node's position within the graph from a different perspective. For example, *degree* is the number of neighbors of the node and *harmonic centrality* measures how important/influential a node is. They also include two community features corresponding to the ID of the node's communities computed respectively with the greedy modularity community detection method of Clauset et al. (2004) and the label propagation algorithm of Cordasco and Gargano (2010). These two methods detect communities of nodes such that there are many links within the communities and only a few between them.

3.3 GLP architecture

Figure 3 displays the architecture of our GLP model; white nodes are for the source (= training) languages and green nodes for the target languages. The model has two parts: the GNN-based *encoder* turns the alignment graph into node representations and the *classifier* learns to label nodes based on these representations. The network is trained to reproduce POS tags for each source node; it is then used to predict the unknown tags for target nodes.

The encoder has two GATConv layers (Veličković et al., 2018): given a graph with M nodes represented as $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M$, with respective neighborhoods $\mathcal{N}(1), \mathcal{N}(2), ..., \mathcal{N}(M)$, a GATConv layer computes a new representation $\mathbf{x}'_i$ for each node as:

$$\mathbf{x}'_i = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{i,j} \mathbf{W} \mathbf{x}_j, \qquad (1)$$

where $\mathbf{W}$ is a learnable weight matrix. $\alpha_{i,j}$ measures how much node i "attends" to node j as follows:

$$\alpha_{i,j} = \frac{\exp\left(g\left(\mathbf{a}^\top [\mathbf{W}\mathbf{x}_i \,\|\, \mathbf{W}\mathbf{x}_j]\right)\right)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp\left(g\left(\mathbf{a}^\top [\mathbf{W}\mathbf{x}_i \,\|\, \mathbf{W}\mathbf{x}_k]\right)\right)}$$

where $\|$ stands for concatenation, g is the LeakyReLU (Maas et al., 2013), and $\mathbf{a}$ is a weight vector. As neighborhoods only use alignment links, the representation of a node is only influenced by nodes in other languages. Also note that both source and target nodes are fed to the encoder.

We train two GLP models: GLP-Base (GLP-B) and GLP-SelfLearning (GLP-SL). The first one is the basic GNN architecture. It tags a token based on the other languages only, i.e. it makes no use of the sequence information of the current token in its own language. The second additionally employs self-learning and is given access to the local context of each token in its own language.

GLP-B uses a multi-layer perceptron as classifier. We feed the node representations to the classifier and train the model end-to-end. We can only do this for source nodes since we have no training data for the target languages.

GLP-SL additionally employs self-learning and a better classifier. Self-learning takes advantage of node labels predicted by GLP-B in the first step: when the prediction confidence exceeds a threshold γ, these labels are deemed correct and the corresponding nodes are considered when training the classifier. GLP-SL uses a Transformer architecture to predict POS tags. The Transformer input consists of all translations of a sentence, where words are represented as GNN node embeddings. Each embedding is the concatenation of input (x_i) and output representations (x'_i) of the corresponding node in the GNN. In addition to the information available from neighbor nodes in *other* languages, the Transformer can attend to other words of the sentence in the *same* language, some of which may already be (automatically) labeled. This is very different from the training of GLP-B, where the POS of words of the same language were either all known (for source languages) or all unknown (for target languages), and explains why we resorted to a simpler classifier in the first stage.

Similarly to Eskander et al. (2020) and Agić et al. (2016), GLP-SL uses type-level information: for each word type, we create a tag distribution by accumulating counts of the number of times each tag was assigned. For source words, we use the training data to estimate the distribution. For target words, we use the predictions of GLP-B on PBC.

3.4 Neural POS tagger

We use the noisy labeled data, generated by GLP-B or GLP-SL, to train monolingual neural POS taggers. Each model is a Bi-LSTM (Bidirectional Long Short-Term Memory, (Hochreiter and

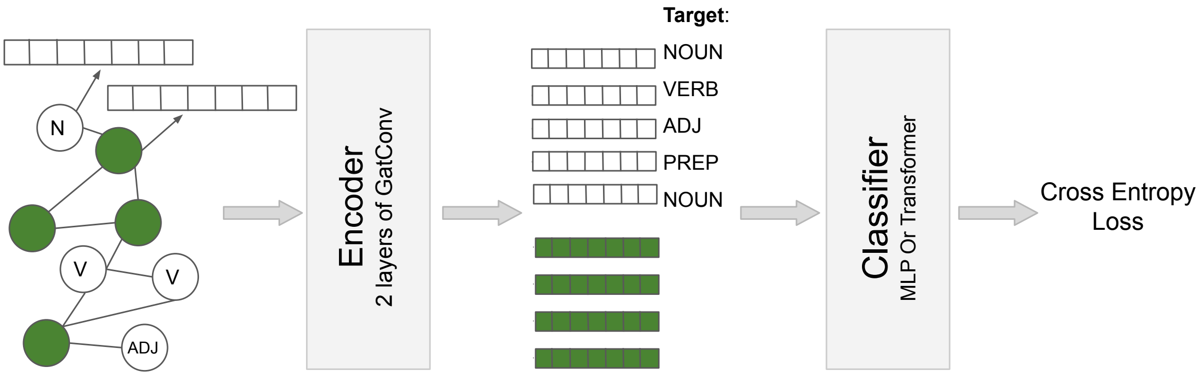


Figure 3: The architecture of GLP (Graph Label Projector). Source nodes are in white, target nodes in green. For training, we first feed the alignment graph of a sentence to the encoder to compute a representation for each node. Next we feed the representations of the source nodes to the classifier. The training objective is cross entropy on prediction of POS tags. Note that we know the POS tags of the source nodes. After training, the model can generalize the POS tag prediction to target nodes.

Schmidhuber, 1997) with XLM-RoBERTa embeddings (Conneau et al., 2020). The input is a sentence labeled by GLP-B or GLP-SL. A token is assigned the NULL tag in case of missing labels. It is then ignored (i.e., masked) when computing the cross-entropy loss. To avoid predicting NULL, we set the corresponding output cell in the softmax to $-\infty$, similarly to Eskander et al. (2020).

	Lang	ISO	Family	# verses
Training languages	Arabic	arb	Afro-Asiatic, Semitic	31173
	Chinese	zho	Sino-Tibetan, Sinitic	31157
	Danish	dan	Indo-European, Germanic	31173
	English	eng	Indo-European	31099
	Finnish	fin	Uralic, Finnic	30200
	French	fra	Indo-European, Romance	31173
	German	deu	Indo-European, Germanic	31173
	Irish	gle	Indo-European, Celtic	34957
	Italian	ita	Indo-European, Romance	35377
	Polish	Pol	Indo-European, Slavic	31157
	Russian	rus	Indo-European, Slavic	31173
	Spanish	spa	Indo-European, Romance	31157
	Swedish	swe	Indo-European, Germanic	31157
	Tamil	tam	Dravidian, Southern Dravidian	7942
	Urdu	urd	Indo-European, Indic	7046
Dev languages	Czech	ces	Indo-European, Slavic	31157
	Greek	ell	Indo-European, Greek	31173
	Hebrew	heb	Afro-Asiatic, Semitic	23174
	Hungarian	hun	Uralic, Ugric	31157
Test languages	Afrikaans	afr	Indo-European, Germanic	31157
	Amharic	amh	Afro-Asiatic, Semitic	7942
	Basque	eus	Basque, Basque	7958
	Belarusian	bel	Indo-European, Slavic	7958
	Bulgarian	bul	Indo-European, Slavic	31173
	Hindi	hin	Indo-European, Indic	7952
	Indonesian	ind	Austronesian, Malayo-Sumbawan	31157
	Lithuanian	lit	Indo-European, Baltic	31149
	Marathi	mar	Indo-European, Indic	7947
	Persian	pes	Indo-European, Iranian	7931
	Portuguese	pos	Indo-European, Romance	31157
	Telugu	tel	Dravidian, South-Central Dravidian	31163
	Turkish	tur	Altaic, Turkic	31157
	Bambara	bam	Mande, Western Mande	7958
	Erzya	myv	Uralic, Mordvin	7958
	Manx	glv	Indo-European, Celtic	3994
	Yoruba	yor	Niger-Congo, Defoid	30819

Table 1: Language family and number of verses in PBC for training, dev, and test languages in our experiments.

4 Experimental setup

Table 1 gives our split of languages into training (15), development (4) and test (17) sets. The training set contains the source languages used for the transfer, while the development set languages are used as targets for parameter tuning. Training and test languages represent diverse language families and diverse availability. Note that training and dev languages are high-resource while test languages are low-resource. For most of the test languages, there are fewer than 8000 verses available in the Parallel Bible Corpus;[4] for Manx, fewer than 4000. We evaluate POS tagging performance on the Universal Dependencies (UD) (Zeman et al., 2019) test sets. As UD and PBC tokenizations differ, we further adopt the following rule: if a PBC token corresponds to a sequence of several UD tokens, we replace the sequence with the original word, tagged with the tag of the UD token in the sequence that is highest in the dependency tree (cf. (Agić et al., 2016)). To tag the high-resource training and dev languages, we use Stanza (Qi et al., 2020),[5] a state-of-the-art NLP Python library. We create word alignments using Eflomal (Östling and Tiedemann, 2016),[6] a high-quality statistical word aligner, with the "intersection" symmetrization heuristic. Other than parallel data, Eflomal does not need any supervision signal; we can thus use it for any language pair in PBC. Details on models' hyperparameters are in Appendix A.3. All tagging results reported below are averages over three runs of the neural

[4] Bible versions are described in Appendix A.1.

[5] https://stanfordnlp.github.io/stanza/

[6] 7github.com/robert/eflomal

with XLM-R													without XLM-R			
afr	amh	eus	bul	hin	ind	lit	pes	por	tel	tur	bel	mar	bam	myv	glv	yor
87.7	82.4	70.9	90.1	81.8	85.3	85.7	81.8	89.2	83.8	80.1	85.9	87.9	65.4	64.4	63.9	59.9

Table 2: Accuracy on UD v2.10 test for GLP-SL when transferring from all training source languages (i.e., GLP-SL-All). See the other tables for comparison with prior work, which uses older versions of UD.

POS tagger model.

5 Results

We evaluate GLP on 17 test languages from different families, resource availabilities, and scripts, on Universal Dependencies v2.10, the latest version (see details in Appendix A.2). Our results are in Table 2. For the four languages not supported by XLM-R, static embeddings are used (see §3.2) during the training in the GNN part (GLP-SL), and XLM-R embeddings in the neural POS tagger model.[7] The best performance, > 89, is obtained for Bulgarian and Portuguese. All scores with XLM-R are above 80, except for Basque. This is probably because no language from the same family appears in the training set. Similarly, Turkish has the lowest performance among the other test languages. Scores without XLM-R are overall lower, yet competitive, showing that our projection method also works for very low-resource languages. Prior work has used older versions of UD . We now compare against each baseline, evaluating on the relevant version of UD in each case.

5.1 Annotation projection-based baselines

In this section, we compare with the unsupervised SOTA in cross-lingual POS tagging via annotation projection: ESKANDER (Eskander et al., 2020), AGIC (Agić et al., 2016) and BUYS (Buys and Botha, 2016b) as well as EFLOMAL. We also compare with a semi-supervised SOTA method that uses rapid annotation in addition to cross-lingual projection: CTRL (Cotterell and Heigold, 2017).

5.1.1 Fully unsupervised baselines

EFLOMAL is a simple projection method using alignment links followed by majority voting, similar to early annotation projection methods (Agić et al., 2015b; Fossum and Abney, 2005). We first align all target sentences with the corresponding sentences in all training languages with Eflomal (Östling and Tiedemann, 2016). Each target word is then tagged with the most common tag in the aligned source words. The annotation projection method ESKANDER (Eskander et al., 2020) uses alignment links and token and type constraints to project tags from source to target. The neural POS tagger features include XLM-R embeddings, affix embeddings, and word clusters created on PBC and Wikipedia of the target languages. Table 3 compares EFLOMAL, ESKANDER and GLP. In this table -Eng stands for when only English is used as the source language in GLP and -All stands for when all training languages are used (see §6.1). GLP outperforms both baselines in all cases but Indonesian, where ESKANDER is 0.7 points better. However, they tune their hyperparameters on this language using dev data while we only tune them on dev languages. Compared to ESKANDER, we use a simpler neural POS tagger and less resources, as we do not use affix embeddings nor word clusters. Our initial experiments indicated that word clusters were not helping in our setup. The higher quality of the annotated data created by GLP may already contain the information provided by word clusters.

Table 4 compares AGIC, BUYS, CTRL, and GLP-SL. AGIC (Agić et al., 2016) is a cross-lingual POS tagger for low-resource languages based on PBC excerpts and translations of the Watchtower.[8] BUYS (Buys and Botha, 2016b) extends previous approaches for projecting POS tags using bitexts to infer constraints on the possible tags for a given word type or token.

Table 4 shows that GLP outperforms AGIC and BUYS, except for Portuguese (BUYS), where our results are slightly below. BUYS projects from Spanish, which is closely related to Portuguese. Eskander et al. (2020) showed that it can be advantageous to transfer only from one closely related language as opposed to a mix of close and distant languages. Note that BUYS performance for Portuguese drops down to 84.3 when transferring from

[7] XLM-R embeddings are used even for languages unseen during its pretraining as they improve performance. This is probably due to the fact that some words (e.g., names) can be well represented even for an unseen language.

[8] Obtained by crawling `http://wol.jw.org`

	afr	amh	eus	bul	hin	ind	lit	pes	por	tel	tur	AVG	bel	mar	AVG
EFLOMAL-Eng	73.7	74.9	60.4	78.9	58.1	72.4	80.3	59.2	74.1	77.5	67.6	70.6	76.2	73.2	71.3
EFLOMAL-All	83.9	79.3	64.5	85.0	68.1	78.4	82.8	68.6	83.8	77.1	74.8	76.9	79.6	77.8	77.2
ESKANDER-Eng	86.9	75.3	67.3	85.6	73.9	**84.1**	80.9	77.2	86.1	80.0	74.3	79.2			
ESKANDER-All	89.3	79.3	67.1	88.2	72.8	83.0	82.5	77.3	87.8	77.1	74.6	79.9			
GLP-B-Eng	86.6	81.9	67.5	85.7	76.8	82.7	81.1	76.2	87.6	82.5	76.4	80.4	80.0	82.3	80.6
GLP-SL-Eng	84.4	81.9	68.6	84.0	75.8	81.3	81.0	73.5	86.4	80.6	75.8	79.4	75.1	81.5	79.2
GLP-B-All	**89.7**	**83.6**	67.4	**89.7**	79.9	82.8	**85.9**	79.6	87.7	81.4	**80.3**	82.5	87.9	83.2	83.0
GLP-SL-All	87.5	82.9	**70.6**	**89.7**	**81.9**	83.4	85.8	**81.9**	**89.6**	**83.7**	78.4	**83.2**	**88.8**	**88.4**	**84.0**

Table 3: Accuracy on UD v2.5 test for EFLOMAL, ESKANDER (Eskander et al., 2020) and GLP. "-Eng": transfer from English only. "-All": transfer from all training languages (see Eskander et al. (2020) and Table 1). Bold: best score for each language.

	Target	AGIC		GLP-SL-All
v1.2	bul	70.0	mul	**86.1**
	hin	50.5	mul	**79.0**
	ind	75.5	mul	**79.5**
	pes	33.7	mul	**75.2**
	por	84.2	mul	**87.7**
	Target	BUYS		GLP-SL-All
v1.2	bul	81.8	eng	**86.1**
	por	**88.0**	esp	87.7
	Target	CTRL		GLP-SL-All
v2.0	Bul	68.8	rus-100	**89.3**
	Bul	83.1	rus-1000	**89.3**
	Por	81.8	esp-100	**90.1**
	Por	88.9	esp-1000	**90.1**

Table 4: Accuracy on UD test for AGIC (Agić et al., 2016), BUYS (Buys and Botha, 2016b), CTRL (Cotterell and Heigold, 2017) and GLP-SL. We also report the source language or "mul" for multilingual, and for CTRL, the number of the supervision tokens.

English. BUYS also uses Europarl[9] with up to 2M tokens which is closer in domain to UD than PBC. Thus, compared to BUYS, the parallel data we use are smaller, and from a more distant domain.

5.1.2 Semisupervised baseline

CTRL (Cotterell and Heigold, 2017) is a character-level recurrent neural network for multi-task cross-lingual transfer of morphological taggers. Their experiments include small sets of 100 and 1000 annotated target tokens. The bottom part of Table 4 shows that GLP-SL outperforms CTRL despite being fully unsupervised.

5.2 Zero-shot baselines

Cross-lingual projection is also possible thanks to multilingual pretrained language models (PLMs). A PLM is first fine-tuned to POS tagging on source languages and then used to infer tags for target languages. While this approach performs well for some languages without requiring any parallel data, its performance tends to be poor for low-resource languages (Hu et al., 2021). Joshi et al. (2020) cluster languages into six groups based on the amount of available unlabeled and labeled data that exists for them. Groups 1 and 2 consist of languages such as Manx and Yoruba with the least amount of available data, while group 5 contains languages like English and Spanish with the largest amount of available monolingual and labeled data. We compare our approach with three baselines using test languages from groups 1 and 2.

mBERT based baselines: Kondratyuk and Straka (2019) use the zero-shot approach with multilingual BERT (Devlin et al., 2019) as PLM. We train our POS taggers using mBERT (instead of XLM-R) embeddings for a fair comparison. Table 5 displays the results for the low-resource languages in group 1 and 2, which are also reported in the compared work. GLP-SL outperforms zero-shot in all cases by at least 12 percentage points. This result suggests that annotation projection using GLP is more effective than using multilingual representations for truly low-resource languages (i.e., languages from the first two groups of Joshi et al. (2020)). To create proper representations for a language, PLMs require a huge amount of monolingual data that is not available for many languages. As Table 5 suggests, due to poor representations, zero-shot transfer to these languages is also poor. However, we were able to successfully exploit the Bible's parallel data in GLP for the benefit of these languages.

XLM-R based baselines: Ebrahimi and Kann (2021) continue pretraining PLMs on PBC and show that this boosts performance for languages unseen during the initial pretraining. Wang et al.

[9] http://www.statmt.org/europarl/

	bam	myv	yor
Kondratyuk and Straka (2019)	30.9	46.7	50.9
GLP-SL-ALL	**65.5**	**64.6**	**63.3**

Table 5: POS tagging accuracy on UD v2.3 test for zero-shot mBERT and GLP-SL using mBERT embeddings.

	bam	myv	glv
Ebrahimi and Kann (2021)	60.5	66.6	59.7
Wang et al. (2022)	69.4	74.3	68.8
GLP-SL-ALL + wang-before	**71.1**	78.9	70.1
GLP-SL-ALL + wang-after	70.2	**80.6**	**70.7**

Table 6: Accuracy on UD v2.5 test for two baselines and for our method combined with (Wang et al., 2022)'s XLM-R models before and after finetuning on the POS tagging task. (“glv” accuracy is on v2.7.)

(2022) adapt PLMs to languages with little monolingual data using various sources of data including PanLex lexicons,[10] translations of English Wikipedia to target languages and the JHU Bible corpus (McCarthy et al., 2020). These approaches are in fact complementary to GLP: we can equip GLP with better multilingual representations to further improve our results based on standard XLM-R. This is reflected in Table 6, where we report results for zero-shot baselines and combinations based on Wang et al. (2022)'s improved XLM-R embeddings (instead of standard XLM-R) to represent tokens for the POS tagger. We see that these combinations lead to large performance improvements, establishing new SOTA results.

6 Analysis

6.1 Ablation study

We conduct an ablation study to better understand what benefits our model.

“Eng” vs “All” Previous works highlighted the importance of a diverse set of source languages for cross-lingual transfer (Lin et al., 2019; Turc et al., 2021). The last four lines of Table 3 report GLP-B and GLP-SL results when transferring from English (i.e., using English as the only source), and when transferring from the full set of source languages (see Table 1). The transfer from English has lower performance than from all languages (except for a decrease from 67.5 to 67.4 for Basque/GLP-B). This means that our projection method does benefit from more data and from the rich information present in the diversity of source languages.

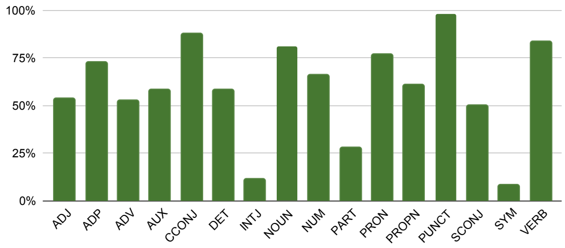


Figure 4: Average per tag accuracy of our GLP sets with respect to the “silver” reference.

GLP-B vs GLP-SL Table 3 reports results when training the neural POS tagger on GLP-B data and on GLP-SL data. GLP-B performs better than GLP-SL for four languages: Afrikaans, Lithuanian, Portuguese, and Turkish; but the performance difference is small (1.2 percentage points difference on average). In eight out of thirteen languages, GLP-SL gives better results (2.3 percentage points difference on average). This shows that the transformer architecture and the self-learning strategy are effective for most languages.

Contextualized vs. Static embeddings Our GLP models use XLM-R embeddings for languages for which they are available, otherwise static embeddings (see §3.2). In order to understand their usefulness in the transfer process, we compare with the performance obtained when static embeddings are used by GLP-SL. Results reported in Appendix B show an average improvement of 3 percentage points when XLM-R embeddings are used. The largest differences ($> 5\%$) are observed for Hindi, Persian, and Marathi. However, for the four languages not supported by XLM-R, the POS tagging accuracy is substantially lower when using contextualized embeddings compared to static embeddings (16.6 points drop on average).

6.2 Quality of artificial training sets

In order to evaluate the quality of the training sets generated by GLP-SL (“GLP sets”), we create a “silver” reference and compute the accuracy of GLP sets with respect to it. To build the silver reference, we annotate the training sets with the Stanza POS tagger for the languages for which it is available (12 out of 17). We obtain an average accuracy of 78.7, with Belarusian being the best and Basque the worst. The best predicted tokens are punctua-

[10] https://panlex.org/snapshot/

tion marks, coordinating conjunctions, and verbs, while the worst ones are symbols, interjections, and particles (see Figure 4). The high accuracy of 78.7 illustrates the ability of GLP-SL to successfully project annotations from high to low-resource languages.

7 Conclusion and future work

We presented GLP, a novel method for transferring labels from high-resource source to low-resource target languages, based on a formalization of annotation projection as graph-based label propagation. We exploited the Parallel Bible Corpus and showed that reasonably accurate POS taggers can be bootstrapped from projected labels. Since we do not use PBC-specific or language-specific features, GLP is in principle applicable to the more than 1000 languages of PBC and to any other multiparallel corpus.

One direction for the future is to employ a similar model to transfer higher-level structures such as dependency trees. Since our method works with graph structures, one might be able to project dependency trees effectively. We could also extend our projection method to other tagging tasks like named entity recognition – although this requires using other parallel corpora to mitigate the domain shift problem of such a task. Another line for future work is to study the best combinations of source languages to transfer to any target language.

Limitations

Our method is evaluated on 17 languages carefully chosen to be from different families and scripts. However, we don't consider the other languages (more than 1000) in PBC due to computational constraints and lack of test sets.

A limitation of the GLP is that training over a MAG (multilingual alignment graph) created for all PBC languages requires a prohibitively large amount of resources, and based on our experiments, if we use a larger number of target languages at the same time, the performance will likely drop. Therefore one has to process languages in smaller batches (in our case, 36 languages). Accordingly, to cover all PBC subcorpora, $1341/36 = 38$ GLP models should in principle be trained.

Ethic statement

Our work is based on the Parallel Bible Corpus of Mayer and Cysouw (2014) that consists of Bible verses and is tested on the Universal Dependency treebanks (Zeman et al., 2019), an ensemble of different data sources. We would like to clarify that we treat the data simply as a multiparallel corpus, and the content does not necessarily reflect the opinions of the authors nor of the institutions funding the authors.

Acknowledgements

We would like to thank Sebastian Ruder for his kind and thoughtful suggestions on this work.

This work was funded by the European Research Council (grant #740516) and the German Federal Ministry of Education and Research (BMBF, grant #01IS18036A).

References

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015a. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272, Beijing, China. Association for Computational Linguistics.

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015b. If all you have is a bit of the bible: Learning pos taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272.

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Paolo Boldi and Sebastiano Vigna. 2014. Axioms for centrality. *Internet Mathematics*, 10(3-4):222–262.

Ulrik Brandes. 2001. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–480.

Jan Buys and Jan A. Botha. 2016a. Cross-lingual morphological tagging for low-resource languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1954–1964, Berlin, Germany. Association for Computational Linguistics.

Jan Buys and Jan A Botha. 2016b. Cross-lingual morphological tagging for low-resource languages. *arXiv preprint arXiv:1606.04279*.

Aaron Clauset, Mark EJ Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Gennaro Cordasco and Luisa Gargano. 2010. Community detection via semi-synchronous label propagation algorithms. In *2010 IEEE international workshop on: business applications of social network analysis (BASNA)*, pages 1–8. IEEE.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. 2018. Embedding learning through multilingual concept induction. *arXiv preprint arXiv:1801.06807*.

Chi Thang Duong, Thanh Dat Hoang, Ha The Hien Dang, Quoc Viet Hung Nguyen, and Karl Aberer. 2019. On node features for graph neural networks. *arXiv preprint arXiv:1911.08795*.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4555–4567, Online. Association for Computational Linguistics.

Ramy Eskander, Smaranda Muresan, and Michael Collins. 2020. Unsupervised cross-lingual part-of-speech tagging for truly low-resource scenarios. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4820–4831, Online. Association for Computational Linguistics.

Meng Fang and Trevor Cohn. 2016. Learning when to trust distant supervision: An application to low-resource POS tagging using cross-lingual projection. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 178–186, Berlin, Germany. Association for Computational Linguistics.

Victoria Fossum and Steven Abney. 2005. Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In *Second International Joint Conference on Natural Language Processing: Full Papers*.

Linton C Freeman. 1978. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239.

Kuzman Ganchev and Dipanjan Das. 2013. Cross-lingual discriminative learning of sequence models with posterior regularization. In *Proceedings of the*

2013 Conference on Empirical Methods in Natural Language Processing, pages 1996–2006, Seattle, Washington, USA. Association for Computational Linguistics.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147, Atlanta, Georgia. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig. 2021. Explicit alignment objectives for multilingual bidirectional encoders. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3633–3643, Online. Association for Computational Linguistics.

Ayyoob Imani, Lütfi Kerem Senel, Masoud Jalili Sabet, François Yvon, and Hinrich Schuetze. 2022. Graph neural networks for multiparallel word alignment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1384–1396, Dublin, Ireland. Association for Computational Linguistics.

Ayyoob ImaniGooghari, Masoud Jalili Sabet, Philipp Dufter, Michael Cysou, and Hinrich Schütze. 2021. ParCourE: A parallel corpus explorer for a massively multilingual corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 63–72, Online. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. 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)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Omer Levy, Anders Søgaard, and Yoav Goldberg. 2017. A strong baseline for learning cross-lingual word embeddings from sentence alignments. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 765–774, Valencia, Spain. Association for Computational Linguistics.

Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223.

Shen Li, João Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1389–1398, Jeju Island, Korea. Association for Computational Linguistics.

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

Zhiyuan Liu and Jie Zhou. 2020. Introduction to graph neural networks. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(2):1–127.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Ma- chine Learning, Atlanta, Georgia, USA*.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland. European Language Resources Association (ELRA).

Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. 2020. The Johns Hopkins University Bible corpus: 1600+

tongues for typological exploration. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France. European Language Resources Association.

Mark EJ Newman. 2001. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1):016132.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072.

Barbara Plank and Željko Agić. 2018. Distant supervision from disparate sources for low-resource part-of-speech tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 614–620, Brussels, Belgium. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 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*, pages 101–108, Online. Association for Computational Linguistics.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Silvia Severini, Ayyoob Imani, Philipp Dufter, and Hinrich Schütze. 2022. Towards a broad coverage named entity resource: A data-efficient approach for many diverse languages. *arXiv preprint arXiv:2201.12219*.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical bert models for sequence labeling. 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)*, pages 3632–3636.

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of English in zero-shot cross-lingual transfer. *arXiv preprint arXiv:2106.16171*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. 2014. Cross-lingual part-of-speech tagging through ambiguous learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1779–1785.

Yongji Wu, Defu Lian, Yiheng Xu, Le Wu, and Enhong Chen. 2020. Graph convolutional networks with markov random field reasoning for social spammer detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1054–1061.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Daniel Zeman, Joakim Nivre, and Mitchell et al. Abrams. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-state LSTM for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327, Melbourne, Australia. Association for Computational Linguistics.

A Reproducibility details

A.1 Data editions

Table 8 lists the PBC editions used for all the experiments in this paper.

A.2 Universal Dependency tests specification

Table 9 lists the Universal Dependency testsets used in our experiments.

	afr	amh	eus	bul	hin	ind	lit	pes	por	tel	tur	bel	mar	AVG	bam	myv	glv	yor	AVG
with XLM-R	87.7	82.4	70.9	90.1	81.8	85.3	85.7	81.8	89.2	83.8	80.1	85.9	87.9	**84.1**	43.0	55.2	50.0	39.0	46.8
without XLM-R	88.4	82.8	72.7	89.3	73.7	80.2	83.9	71.3	85.0	80.1	77.8	85.2	82.0	81.0	65.4	64.4	63.9	59.9	**63.4**

Table 7: Accuracy on UD v2.10 for GLP-SL when transferring from all training languages (i.e., GLP-SL-All) with and without using XLM-R for the transfer in GLP-SL.

Lang	Edition	Lang	Edition
Arabic	arb-x-bible	Hungarian	hun-x-bible-newworld
Chinese	zho-x-bible-newworld	Afrikansc	afr-x-bible-newworld
Danish	dan-x-bible-newworld	Amharic	amh-x-bible-newworld
English*	eng-x-bible-mixed	Basque	eus-x-bible-navarrolabourdin
Finnish*	fin-x-bible-helfi	Belarusian	bel-x-bible-bokun
French	fra-x-bible-louissegond	Bulgarian	bul-x-bible-newworld
German	deu-x-bible-bolsinger	Hindi	hin-x-bible-bsi
Irish	gle-x-bible	Indonesian	ind-x-bible-newworld
Italian	ita-x-bible-2009	Lithuanian	lit-x-bible-ecumenical
Polish	pol-x-bible-newworld	Marathi	mar-x-bible
Russian	rus-x-bible-newworld	Persian	pes-x-bible-newmillennium2011
Spanish	spa-x-bible-newworld	Portuguese	por-x-bible-newworld1996
Swedish	swe-x-bible-newworld	Telugu	tel-x-bible
Tamil	tam-x-bible-newworld	Turkish	tur-x-bible-newworld
Urdu	urd-x-bible-2007	Bambara	bam-x-bible
Czech	ces-x-bible-newworld	Erzya	myv-x-bible
Greek	ell-x-bible-newworld	Manx	glv-x-bible
Hebrew*	heb-x-bible-helfi	Yoruba	yor-x-bible-2010

Table 8: PBC editions for all used languages. *Edition from Imani et al. (2022).

Lang	Test
Afrikaans	af_afribooms-ud-test
Amharic	am_att-ud-test
Basque	eu_bdt-ud-test
Belarusian	be_hse-ud-test
Bulgarian	bg_btb-ud-test
Hindi	hi_hdtb-ud-test
Ind	id_gsd-ud-test
Lithuanian	lt_alksnis-ud-test
Marathi	mr_ufal-ud-test.
Persian	fa_seraji-ud-test
Portuguese	pt_bosque-ud-test
Telugu	te_mtg-ud-test
Turkish	tr_imst-ud-test
Bambara	bm_crb-ud-test
Erzya	myv_jr-ud-test
Manx	gv_cadhan-ud-test
Yoruba	yo_ytb-ud-test

Table 9: Universal Dependency test sets used in our experiments.

A.3 Models parameters

GLP The GLP is implemented using the PyTorch geometric library.[11] All hyperparameters are tuned on the dev set. GLP-B has 2 layers of MLP of size 2048 while GLP-SL uses four layers of transformer with hidden size 2048 and 16 attention heads. Although we didn't observe a difference between different sizes from 512 to 2048. We tuned the learning rate, batch size, and γ (the self-learning threshold) over the validation languages. In GLP-B learning rate and batch size are respectively 0.001, 8, and in GLP-SL 0.00001, and 32. In general, when using XLM-R embeddings, the model has higher confidence, so the γ parameter is set to 0.95 when not using XLM-R embeddings and 0.98 when using XLM-R embeddings. The whole model needs about $16GB$ of GPU memory. GLP-B takes about 2 hours to train and GLP-SL about 12 hours. We used early stopping with patience of 8 for both GLP-B and GLP-SL.

Neural POS tagger We run our method on up to 48 cores of Intel(R) Xeon(R) CPU E7-8857 v2 with 1TB memory and a single GeForce GTX 1080 GPU with 8GB memory. The POS tagger uses the Flair framework (Akbik et al., 2019) and SequenceTagger model with 128 hidden size, the "xlm-roberta-base" embeddings, and AdamW optimizer Loshchilov and Hutter (2018). The hyperparameters, including the fixed number of epochs (15) are tuned using the UD development sets of the development languages. Each Neural POS tagger was trained in less than 30 minutes.

B Contextualized vs. Static embeddings

Table 7 shows results obtained with our GLP-SL with and without using XLM-R embeddings for projection. Note that the final neural POS tagger models always use XLM-R embeddings, even for languages unseen during XLM-R pretraining.

[11] https://pytorch-geometric.readthedocs.io/en/latest/

Bibliography

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015. If all you have is a bit of the bible: Learning pos taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272.

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Željko Agić and Ivan Vulić. 2019. Jw300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

KK Akhil, R Rajimol, and VS Anoop. 2020. Parts-of-speech tagging for malayalam using deep learning techniques. *International Journal of Information Technology*, 12(3):741–748.

Jay Alammar. 2018. The illustrated transformer. *The Illustrated Transformer–Jay Alammar–Visualizing Machine Learning One Concept at a Time*, 27.

Jacob Andreas and Dan Klein. 2014. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 822–827, Baltimore, Maryland. Association for Computational Linguistics.

Monika Arora and Vineet Kansal. 2019. Character level embedding with deep convolutional neural network for text normalization of unstructured data for twitter sentiment analysis. *Social Network Analysis and Mining*, 9(1):1–14.

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

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. 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.

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

Pranjal Awasthi, Delip Rao, and Balaraman Ravindran. 2006. Part of speech tagging and chunking with hmm and crf. *Proceedings of NLP Association of India (NLPAI) Machine Learning Contest 2006*.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Luisa Bentivogli, Pamela Forner, and Emanuele Pianta. 2004. Evaluating cross-language annotation transfer in the MultiSemCor corpus. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 364–370, Geneva, Switzerland. COLING.

Sara Besharati, Hadi Veisi, Ali Darzi, and Seyed Habib Hosseini Saravani. 2021. A hybrid statistical and deep learning based technique for persian part of speech tagging. *Iran Journal of Computer Science*, 4(1):35–43.

Damián Blasi, Antonios Anastasopoulos, and Graham Neubig. 2022. Systematic inequalities in language technology performance across the world's languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5505.

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.

Fabienne Braune, Viktor Hangya, Tobias Eder, and Alexander Fraser. 2018. Evaluating bilingual word embeddings on the long tail. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 188–193.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Yun-Chuang Chiao and Pierre Zweigenbaum. 2002. Looking for candidate translational equivalents in specialized, comparable corpora. In *COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*.

Kyunghyun Cho, B van Merrienboer, Caglar Gulcehre, F Bougares, H Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069.

Paula Czarnowska, Sebastian Ruder, Édouard Grave, Ryan Cotterell, and Ann Copestake. 2019. Don't forget the long tail! a comprehensive analysis of morphological generalization in bilingual lexicon induction. 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)*, pages 974–983.

Tirthankar Dasgupta, Manjira Sinha, and Anupam Basu. 2013. A joint source channel model for the english to bengali back transliteration. In *Mining intelligence and knowledge exploration*, pages 751–760. Springer.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Philipp Dufter, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. 2018. Embedding learning through multilingual concept induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1520–1530.

Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an unsupervised transliteration model into statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 148–153.

David M. Eberhard, F. Simons Gary, and Charles. D. Fennig (eds.). 2020. Ethnologue: Languages of the world. *23rd edition. SIL International.*

Tobias Eder, Viktor Hangya, and Alexander Fraser. 2021. Anchor-based bilingual word embeddings for low-resource languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 227–232.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. Characterbert: Reconciling elmo and bert for word-level open-vocabulary representations from characters. In *International Conference on Computational Linguistics*, pages 6903–6915.

Ramy Eskander, Smaranda Muresan, and Michael Collins. 2020. Unsupervised cross-lingual part-of-speech tagging for truly low-resource scenarios. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4820–4831, Online. Association for Computational Linguistics.

Andrew Finch, Paul Dixon, and Eiichiro Sumita. 2012. Rescoring a phrase-based machine transliteration system with recurrent neural network language models. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, pages 47–51.

Andrew Finch, Keiji Yasuda, Hideo Okuma, Eiichiro Sumita, and Satoshi Nakamura. 2011. A bayesian model of transliteration and its human evaluation when integrated into a machine translation system. *IEICE transactions on Information and Systems*, 94(10):1889–1900.

Victoria Fossum and Steven Abney. 2005. Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In *International Conference on Natural Language Processing*, pages 862–873. Springer.

Jianfeng Gao and Mark Johnson. 2007. A comparison of bayesian estimators for unsupervised hidden markov model pos taggers.

Yingqiang Gao, Nikola I Nikolov, Yuhuang Hu, and Richard HR Hahnloser. 2020. Character-level translation with self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1591–1604.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):1–309.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks*, volume 2, pages 729–734.

Edouard Grave, Armand Joulin, and Quentin Berthet. 2019. Unsupervised alignment of embeddings with wasserstein procrustes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1880–1890. PMLR.

Ralph Grishman and Beth M Sundheim. 1996. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Benjamin Heinzerling and Michael Strube. 2019. Sequence tagging with contextual and non-contextual subword representations: A multilingual evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 273–291.

Ulf Hermjakob, Kevin Knight, and Hal Daumé III. 2008. Name translation in statistical machine translation-learning when to transliterate. In *Proceedings of ACL-08: HLT*, pages 389–397.

Ulf Hermjakob, Jonathan May, and Kevin Knight. 2018. Out-of-the-box universal Romanization tool uroman. In *Proceedings of ACL 2018, System Demonstrations*, pages 13–18, Melbourne, Australia. Association for Computational Linguistics.

Geoffrey E Hinton. 1984. Distributed representations.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(3):311–325.

Ayyoob Imani, Lütfi Şenel, Masoud Sabet, François Yvon, and Hinrich Schütze. 2022. Graph neural networks for multiparallel word alignment. In *Findings of the ACL*.

Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim, and Grzegorz Kondrak. 2010. Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, pages 39–47.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293.

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2011. Machine transliteration survey. *ACM Computing Surveys (CSUR)*, 43(3):1–46.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294, Melbourne, Australia. Association for Computational Linguistics.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational linguistics-Association for Computational Linguistics (Print)*, 24(4):599–612.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. 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)*, pages 2779–2795.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018a. Word translation without parallel data. In *International Conference on Learning Representations*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *EMNLP*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27:2177–2185.

Bing Li, Yujie He, and Wenjin Xu. 2021. Cross-lingual named entity recognition using parallel corpus: A new approach using xlm-roberta alignment. *arXiv preprint arXiv:2101.11112*.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 159–166.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado. Association for Computational Linguistics.

Khai Mai, Thai-Hoang Pham, Minh Trung Nguyen, Tuan Duc Nguyen, Danushka Bollegala, Ryohei Sasano, and Satoshi Sekine. 2018. An empirical study on fine-grained named entity recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 711–722.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel bible corpus. *Oceania*, 135(273):40.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30.

Helen M Meng, Berlin Chen, Sanjeev Khudanpur, Gina-Anne Levow, Wai-Kit Lo, Douglas Oard, Patrick Schone, Karen Tang, Hsin-min Wang, and Jianqiang Wang. 2004. Mandarin–english information (mei): investigating translingual speech retrieval. *Computer Speech & Language*, 18(2):163–179.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Muhammad Tasnim Mohiuddin, M Saiful Bari, and Shafiq Joty. 2020. Lnmap: Departures from isomorphic assumption in bilingual lexicon induction through non-linear mapping in latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2712–2723.

Molly Moran and Constantine Lignos. 2020. Effective architectures for low resource multilingual named entity transliteration. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 79–86.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mbert is just the beginning: Handling new languages with multilingual language models. In *NAACL-HLT 2021-2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ahn-Khoa Ngo-Ho and François Yvon. 2019. Neural baselines for word alignments. In *International Workshop on Spoken Language Translation*.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 440–447.

Jong-Hoon Oh and Key-Sun Choi. 2002. An english-korean transliteration model using pronunciation and contextual rules. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Jong-Hoon Oh and Key-Sun Choi. 2006. An ensemble of transliteration models for information retrieval. *Information processing & management*, 42(4):980–1002.

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

Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Robert Östling, Jörg Tiedemann, et al. 2016. Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics*.

Sebastian Padó and Mirella Lapata. 2005. Cross-linguistic projection of role-semantic information. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 859–866, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 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)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Barbara Plank and Željko Agić. 2018. Distant supervision from disparate sources for low-resource part-of-speech tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 614–620, Brussels, Belgium. Association for Computational Linguistics.

Dinesh Kumar Prabhakar and Sukomal Pal. 2018. Machine transliteration and transliterated text retrieval: a survey. *Sādhanā*, 43(6):1–25.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(sept):2487–2531.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 320–322.

Reinhard Rapp, Pierre Zweigenbaum, and Serge Sharoff. 2020. Overview of the fourth bucc shared task: Bilingual dictionary induction from comparable corpora. In *13th Workshop on Building and Using Comparable Corpora (BUCC)*, pages 6–13.

Mihaela Rosca and Thomas Breuel. 2016. Sequence-to-sequence neural network models for transliteration. *arXiv preprint arXiv:1610.09565*.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Antoine de Saint-Exupéry. 1943. Le petit prince [the little prince]. *Verenigde State van Amerika: Reynal & Hitchkock (US), Gallimard (FR).*

Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 469–477.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

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

Hinrich Schütze. 1992. Dimensions of meaning. In *SC*, pages 787–796.

Yan Shao and Joakim Nivre. 2016. Applying neural networks to english-chinese named entity transliteration. In *Proceedings of the sixth named entity workshop*, pages 73–77.

Serge Sharoff, Reinhard Rapp, Pierre Zweigenbaum, and Pascale Fung. 2013. *Building and using comparable corpora*. Springer.

Serge Sharoff, Pierre Zweigenbaum, and Reinhard Rapp. 2015. Bucc shared task: Cross-language document similarity. In *Proceedings of the Eighth Workshop on Building and Using Comparable Corpora*, pages 74–78.

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

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 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)*, pages 778–788.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating names and technical terms in arabic text. In *Computational Approaches to Semitic Languages*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Jörg Tiedemann. 2020. The tatoeba translation challenge–realistic data sets for low resource and multilingual mt. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182.

Kentaro Torisawa et al. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *proceedings of ACL-08: HLT*, pages 407–415.

Chen-Tse Tsai and Dan Roth. 2016. Illinois cross-lingual wikifier: Grounding entities in many languages to the english wikipedia. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 146–150.

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

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*, pages 5998–6008.

Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition*, pages 57–64.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the best of cross-lingual transfer: Evidence from pos tagging with over 100 languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685.

Ivan Vulic and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, volume 2, pages 719–725. ACL; East Stroudsburg, PA.

Yu-Chun Wang, Chun-Kai Wu, and Richard Tzong-Han Tsai. 2015. Ncu iisr english-korean and english-chinese named entity transliteration using different grapheme segmentation approaches. In *Proceedings of the Fifth Named Entity Workshop*, pages 83–87.

Yuxuan Wang, Yutai Hou, Wanxiang Che, and Ting Liu. 2020. From static to dynamic word representations: a survey. *International Journal of Machine Learning and Cybernetics*, 11(7):1611–1630.

Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Chun-Kai Wu, Yu-Chun Wang, and Richard Tzong-Han Tsai. 2012. English-korean named entity transliteration using substring alignment and re-ranking methods. In *Proceedings of the 4th Named Entity Workshop (NEWS) 2012*, pages 57–60.

Qianhui Wu, Zijia Lin, Börje F Karlsson, Biqing Huang, and Jianguang Lou. 2020. Unitrans: Unifying model transfer and data transfer for cross-lingual named entity recognition with unlabeled data. In *IJCAI*.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907.

Winston Wu, Nidhi Vyas, and David Yarowsky. 2018. Creating a translation matrix of the bible's names across 591 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Winston Wu and David Yarowsky. 2018. A comparative study of extremely low-resource transliteration of the world's languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1006–1011.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*.

Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zhong Zhou and Alex Waibel. 2021. Family of origin and family of choice: Massively parallel lexiconized iterative pretraining for severely low resource text-based translation. In *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*, pages 67–80.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1393–1398.