Distributed Representations for Multilingual Language Processing

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

eingereicht von
Philipp Dufter

München, den 9. Dezember 2020

Erstgutachter: *Prof. Dr. Hinrich Schütze*
Zweitgutachter: *Dr. François Yvon*
Drittgutachter: *Prof. Dr. Goran Glavaš*

Tag der Einreichung: 9. Dezember 2020
Tag der mündlichen Prüfung: 28. April 2021

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 9. Dezember 2020

Philipp Dufter

Abstract

Distributed representations are a central element in natural language processing. Units of text such as words, ngrams, or characters are mapped to real-valued vectors so that they can be processed by computational models. Representations trained on large amounts of text, called static word embeddings, have been found to work well across a variety of tasks such as sentiment analysis or named entity recognition. More recently, pretrained language models are used as contextualized representations that have been found to yield even better task performances.

Multilingual representations that are invariant with respect to languages are useful for multiple reasons. Models using those representations would only require training data in one language and still generalize across multiple languages. This is especially useful for languages that exhibit data sparsity. Further, machine translation models can benefit from source and target representations in the same space. Last, knowledge extraction models could not only access English data, but data in any natural language and thus exploit a richer source of knowledge.

Given that several thousand languages exist in the world, the need for multilingual language processing seems evident. However, it is not immediately clear, which properties multilingual embeddings should exhibit, how current multilingual representations work and how they could be improved.

This thesis investigates some of these questions. In the first publication, we explore the boundaries of multilingual representation learning by creating an embedding space across more than one thousand languages. We analyze existing methods and propose concept based embedding learning methods. The second paper investigates differences between creating representations for one thousand languages with little data versus considering few languages with abundant data. In the third publication, we refine a method to obtain interpretable subspaces of embeddings. This method can be used to investigate the workings of multilingual representations. The fourth publication finds that multilingual pretrained language models exhibit a high degree of multilinguality in the sense that high quality word alignments can be easily extracted. The fifth paper investigates reasons why multilingual pretrained language models are multilingual despite lacking any kind of crosslingual supervision during training. Based on our findings we propose a training scheme that leads to improved multilinguality. Last, the sixth paper investigates the use of multilingual pretrained language models as multilingual knowledge bases.

Zusammenfassung

Verteilte Repräsentationen sind ein zentrales Element in der automatischen Verarbeitung natürlicher Sprachen. Funktionen weisen Texteinheiten wie Wörtern, N-Grammen oder Buchstaben reellwertige Vektoren zu, sodass diese von Computern mit Rechenmodellen verarbeitet werden können. Sogenannte statische Wortrepräsentationen, die auf großen Mengen von Text gelernt werden, sind nützlich für Aufgaben wie Sentimentanalyse oder Entitätenerkennung. Kürzlich wurden kontextualisierte Repräsentationen entwickelt. Diese können die genannten Aufgaben noch effektiver lösen.

Multilinguale Repräsentationen, also Repräsentationen, die invariant bezüglich eines Sprachwechsels sind, sind aus mehreren Gründen nützlich. Zum einen müssen Modelle, die eine bestimme Aufgabe wie Entitätenerkennung lösen und diese Repräsentationen verwenden, nur mit Trainingsdaten in einer Sprache trainiert werden. Es ist also ausreichend, dass annotierte Daten in einer Sprache vorliegen. Trotzdem können die Modelle mit Hilfe der multilingualen Repräsentationen Daten, die in einer anderen Sprache vorliegen, verarbeiten. Sprachen, bei denen wenig Textdaten verfügbar sind, können davon besonders profitieren. Zum anderen kann maschinelle Übersetzung mit Hilfe von multilingualen Repräsentationen verbessert werden. Nicht zuletzt können auch Informationsextraktionsmodelle mit Hilfe der Repräsentationen nicht nur englische Daten, sondern Daten in verschiedenen Sprachen verarbeiten und so insgesamt mehr Informationsquellen berücksichtigen.

Da es mehrere tausend Sprachen auf der Welt gibt, ist es vielversprechend, multilinguale Modelle zu entwickeln. Dabei muss spezifiziert werden, welche Eigenschaften multilinguale Modelle haben sollen, wie diese im Detail funktionieren und wie diese verbessert werden können.

Das Ziel der vorliegenden Arbeit ist es, einige dieser Fragen zu untersuchen. In der ersten Veröffentlichung analysieren wir die Grenzen multilingualer Modelle indem wir Repräsentationen für tausende Sprachen erstellen. Wir untersuchen existierende Methoden und entwickeln konzeptbasierte Algorithmen. Das zweite Papier ergründet Unterschiede zwischen dem Lernen von Repräsentationen für mehr als tausend Sprachen mit wenig Daten und für wenige Sprachen mit vielen Daten. In der dritten Veröffentlichung verbessern wir eine Methode, um Repräsentationen interpretierbar zu machen. Das kann nützlich sein, um multilinguale Repräsentationen besser zu verstehen. Wir planen, diese Methode in zukünftigen Arbeiten einzusetzen. Die vierte Arbeit zeigt, dass qualitativ hochwertige Wortalignierungen mit Hilfe von vortrainierten Repräsentationen erstellt werden können. In einer fünften Veröffentlichung untersuchen wir, warum kontextualisierte Wortrepräsentationen multilingual sind, trotz fehlender Anreize wäh-

rend des Lernprozesses. Darauf basierend stellen wir eine Trainingsmethode vor, die zu einem höheren Grad an Multilingualität führt. Im letzten Papier untersuchen wir, ob multilinguale kontextualisierte Repräsentationen Wissen über Entitäten enthalten.

Acknowledgments

This work would not have been possible without strong support of many people. First and foremost, I would like to thank my supervisor Hinrich Schütze. Only your endless patience, unfailing support, and you believing in me made it possible to finish this work. I learned countless things from you and hope that one day I will become as good as you in listening to, understanding, and solving so many problems at once.

Many thanks to François Yvon and Goran Glavaš for reviewing the dissertation in such detail and stimulating a very interesting discussion during the disputation, and to the whole committee for taking the time out of your busy schedules for the examination.

The years at CIS have been fantastic. Besides great excursions and events, I will miss the uncomplicated way of collaborating and the wonderful colleagues. Thanks to Nora Kassner and Martin Schmitt for being great office mates, and for writing papers together while listening to the distant Blasmusik from the nearby beer garden. Thanks to Masoud Jalili Sabet, Mengjie Zhao, Benjamin Roth, Nina Pörner, Ayyoob Imani, Ehsaneddin Asgari, Sheng Liang, Silvia Severini, Antonis Maronikolakis, Alexander Fraser, Helmut Schmid for working on projects together – they were a highlight during my PhD, and to all other colleagues and alumni at CIS – I will miss the countless inspiring discussions, lunches and whiteboard sessions in the kitchen. Thanks to Peggy Hobmaier for help in manoeuvering the administrative jungle and Thomas Schäfer for being an incredible supportive and fast admin.

I gratefully acknowledge a fellowship by the Bavarian Research Institute for Digital Transformation for funding most of my PhD – thanks to the peer group of fellows for incredible fun workshops; funding by the European Research Council (# 740516); and support through a Bosch AI Young Scientist fellowship – many thanks to Heike Adel and Jannik Strötgen.

Friends outside of work were essential for keeping my sane during the past years: Cornelius, Dani, Huaba, Joe, Julian, Moritz, Nelson, Tessa, Timo, and to all others – thanks for unforgettable times. To Fabian: thank you for your companionship – I am looking forward to walk through the next decades together. I could not be at this point without the support of my family: to my parents, sisters, brother-in-laws, nephews, nieces – thanks for always being there for me. Special thanks to Korbinian for countless advice and support – I am so lucky to have such an exceptional brother. Most importantly, I thank Johanna for her endless love and for accompanying me through all times – you are the best that happened to me.

Publications and Declaration of Co-Authorship

Chapter 2 corresponds to the following publication:

> **Philipp Dufter**, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. *Embedding Learning Through Multilingual Concept Induction.* In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1520-1530. 2018.

Hinrich Schütze and I conceived of the original research contributions. I performed most of the implementations and evaluations, except for the character level alignment (Hinrich Schütze) and sentiment analysis (Mengjie Zhao). Martin Schmitt helped with the annotation of the sentiment gold standard. I wrote the initial draft of the article and did most of the subsequent corrections. I regularly discussed this work with my co-authors who assisted me in improving the manuscript.

Chapter 3 corresponds to the following preprint:

> **Philipp Dufter**, Mengjie Zhao, and Hinrich Schütze. *Multilingual Embeddings Jointly Induced from Contexts and Concepts: Simple, Strong and Scalable.* Computing Research Repository, arXiv:1811.00586. 2018.

I conceived of the original research contributions and performed all implementations and evaluations except for the sentiment analysis evaluation (Mengjie Zhao). I wrote the initial draft of the article and did most of the subsequent corrections. I regularly discussed this work with my co-authors who assisted me in improving the draft.

Chapter 4 corresponds to the following publication:

> **Philipp Dufter** and Hinrich Schütze. *Analytical Methods for Interpretable Ultradense Word Embeddings.* In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 1185-1191. 2019.

I conceived of the original research contributions and performed all implementations and evaluations. I wrote the initial draft of the article and did most of the subsequent corrections. I regularly discussed this work with my advisor who assisted me in improving the draft.

Chapter 5 corresponds to the following publication:

> Masoud Jalili Sabet*, **Philipp Dufter***, François Yvon, and Hinrich Schütze. *SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings.* In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): Findings, pp. 1627–1643. 2020. *equal contribution.

I conceived of the original research contributions, implemented a proof-of-concept and created an interactive demo. Most subsequent implementations and evaluations were performed by Masoud Sabet. I wrote the initial draft of the article and did most of the subsequent corrections. I regularly discussed this work with my co-authors who assisted me in improving the draft.

Chapter 6 corresponds to the following publication:

> **Philipp Dufter** and Hinrich Schütze. *Identifying Necessary Elements for BERT's Multilinguality.* In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4423–4437. 2020.

I conceived of the original research contributions and performed all implementations and evaluations. I wrote the initial draft of the article and did most of the subsequent corrections. I regularly discussed this work with my advisor who assisted me in improving the draft.

Chapter 7 corresponds to the following publication:

> Nora Kassner*, **Philipp Dufter***, and Hinrich Schütze. *Multilingual LAMA: Investigating Knowledge in Multilingual Pretrained Language Models.* In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pp. 3250–3258. 2021. *equal contribution.

I conceived of the original research contributions. Nora Kassner implemented the evaluation and performed the experiments, while I implemented the code for translating the dataset. Nora Kassner and I wrote the initial draft of the article and we did most of the subsequent corrections together. We regularly discussed this work with Hinrich Schütze who assisted in improving the draft.

München, den 9. Dezember 2020

Philipp Dufter

Contents

Chapter 1

Introduction

1.1 Motivation

Natural languages are central to human communication. Similarly, language is an intuitive way for humans to interact with computers. Therefore, processing, understanding, and generating natural language text with computational models has been a central part of computer science research since its inception. Some early milestones include the Georgetown experiment from 1954 as described in (Hutchins, 2004), which presented a rule-based machine translation system; a few years before, in 1950, Alan Turing used natural language understanding as a criterion to assess the intelligence of a machine in the Turing test (Turing, 1950).

Processing text with computational models requires a numerical representation. As text data is inherently sequential and potentially arbitrarily long, it is useful to divide a sequence of text into smaller units, such as sentences, words, or characters. While a binary (or symbolic) representation for a unit is easy to define, distributed representations are the preferred way of representing text units: meaning can be encoded implicitly through a similarity metric (Hinton et al., 1990; Schütze, 1992b) and distributed representations have been found to interleave well with statistical models such as logistic regression or neural networks (Hinton et al., 1986; Bengio et al., 2003; Collobert et al., 2011). However, it is not clear how to define meaningful distributed representations for textual units. Creating those is the objective of *representation learning*. Factorizations of the co-occurrence matrices (Schütze, 1992a) have been found to yield meaningful distributed representations in the sense that for example cosine similarity correlates well with semantic similarity. Used as the first layer in neural networks they yield good performance on downstream tasks such as named entity recognition or sentiment analysis (Collobert et al., 2011). More recently, pretrained language models are the preferred way of creating contextualized unit representations.

Processing a single natural language has obvious limitations. The European Union has 24 official languages, India lists 22 languages and overall there are more than 7000 diverse languages in the world (Eberhard et al., 2020). While one could create separate language processing systems and separate distributed representations for each language, there are several arguments for creating *multilingual representations*:

i) Multilingual models require less data, both written text and annotated training data, as they generalize across multiple languages. This is especially useful for languages that exhibit data sparsity and saves annotation resources.

ii) Machine translation can benefit from source and target representations in the same numerical space in terms of improved quality.

iii) Knowledge extraction models could not only access English data but could read any natural language and thus exploit a richer source of knowledge.

iv) From an engineering perspective it is easier to maintain a single multilingual model rather than a large number of monolingual models.

v) Through developing models in different languages one can work against their digital extinction.

On an abstract level, one can argue that natural languages serve similar purposes, namely the communication between humans and the description and interpretation of the physical world and abstract thoughts. This fact lends itself to representing and processing multiple languages in a single system. Note that sometimes in the literature the terms *multilingual* and *crosslingual* are used with different meanings: multilingual refers to systems that can process multiple languages without any cross-language interaction, whereas crosslingual refers to systems that exhibit meaningful connections between languages. In this work, we always refer to the latter and use both terms interchangeably.

1.1.1 Approach

In this work we approach the goal of multilingual processing by researching multilingual distributed representations. There are two reasons for this choice: numerical representations are an integral part of almost every computational model and through reusing pretrained representations large amounts of text data can be leveraged.

We pursue a *language agnostic* approach in the sense that we apply the same processing pipeline to all languages and omit language specific rules. One can argue that this fails to address the diversity of languages: languages exhibit a

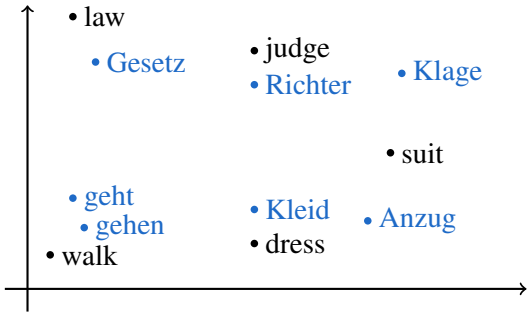


Figure 1.1 – *Example of expectations on multilingual embeddings. Semantically similar words across and within languages are close to each other.*

very different degree of using morphology, e.g., ranging from agglutinative to analytical languages, there are different word order schemes, writing systems and a range of additional linguistic properties. However, we aim at processing hundreds or thousands of languages and language specific processing rules are hard to scale. Thus we focus on researching a single processing pipeline for all considered languages.

1.1.2 Research Questions

Having presented the need for *multilingual distributed representations* there is a range of open questions.

i) *Properties:* It is unclear which properties multilingual representations should exhibit. Figure 1.1 shows an example of how one could imagine multilingual representations. The actual term *multilingual representations* needs to be defined and measures that evaluate the quality of those representations quantitatively are required.

ii) *Limits:* The feasibility and limits of such representations need to be investigated. Natural questions are whether two different languages can be represented in the same numerical space in a meaningful way and whether there is a limit to how many languages can be modeled in the same space.

iii) *Analysis:* The quality and mechanics of existing algorithms need to be understood.

iv) *Improvements:* Insights from the above research questions are used to create representation learners that exhibit a higher degree of multilinguality.

The objective of this work is to address some of these questions.

1.1.3 Outline

We start by introducing the basics of monolingual and multilingual representation learning in Chapter 1. After defining static and contextualized embeddings, we present some popular representation learning algorithms and show how representations can be evaluated. In the first two research papers, which can be found in Chapter 2 and 3, we investigate the limits of multilingual representations by investigating embedding spaces in more than a thousand languages. We derive new embedding learners adapted specifically to this massively multilingual setting. In the third publication in Chapter 4 we improve a method for interpreting distributed representations. We deem this useful for analyzing multilingual representations, which we plan to do in future work. The fourth publication in Chapter 5 analyzes existing multilingual representations and shows that word alignments can be easily obtained from them. In Chapter 6, we analyze the reasons for the multilinguality of an existing model, mBERT, and propose a modification that leads to increased multilinguality. The last paper in Chapter 7 investigates the use of multilingual pretrained language models as multilingual knowledge bases.

By and large, Chapter 2 and 3 address research question ii) by exploring the limits of multilingual representations and research question iv) by proposing concept based embedding learning. Chapter 4 and Chapter 5 contribute to question iii), understanding the quality and mechanics of current algorithms. Chapter 6 targets research question iii) by analyzing multilingual models and iv) by trying to improve multilingual algorithms and in the last article in Chapter 7 we create a resource that can be loosely categorized into research question i).

1.2 Foundations

1.2.1 Notation

Scalar values are lowercase letters $x \in \mathbb{R}$, vectors are boldface lowercase $\mathbf{x} \in \mathbb{R}^d$, and matrices boldface uppercase letters $\mathbf{X} \in \mathbb{R}^{t \times d}$, for positive integers $d, t \in \mathbb{N}^+$. Vectors and matrices can be indexed $(x_i)_{i=1,2\ldots,d} = \mathbf{x}$, $(X_{ij})_{i=1,2\ldots,t,j=1,2,\ldots d} = \mathbf{X}$. The i-th row of $\mathbf{X}$ is denoted as $\mathbf{X}_i \in \mathbb{R}^d$. When we index a matrix with a textual unit instead of an index we refer to the vector that corresponds to this unit, i.e., $\mathbf{X}_{\text{sun}}$ refers to the vector that corresponds to "sun" based on some underlying bijective function that assigns each textual unit a unique integer. The same holds for vectors that are indexed in such a way. The cardinality of a set A is denoted by $|A|$. The power set of A is $\mathcal{P}(A)$. The euclidean norm of a vector $\mathbf{x}$ is $\|\mathbf{x}\|$. $\mathbf{x}^\intercal$ and $\mathbf{X}^\intercal$ are the transposed vector (row-vector) and matrix. For two vectors $\mathbf{x}, \mathbf{y}$ we denote the cosine similarity as $\text{cos-sim}(\mathbf{x}, \mathbf{y}) := \cos(\theta) = \mathbf{x}^\intercal\mathbf{y}/(\|\mathbf{x}\| \, \|\mathbf{y}\|)$. The n-dimensional identity matrix is denoted by $\mathbf{I}_n$. The indicator function $\mathbb{1}\{A\}$ is 1 if and only if the statement A is true and 0 else.

1.2.2 Distributed Representations

We can assume that natural text has a reading direction and can thus be interpreted as sequential data. Therefore, we denote text data as $(u_1, u_2, \ldots, u_t)$ where u_i is some unit of text. We call the set of distinct units the vocabulary $V = \{v_1, v_2, \ldots, v_n\}$. Common choices for units are shown in Figure 1.2. Two units are identical if they consist of an identical sequence of unicode code points. In order to process units of text computationally, they need to be represented numerically. An example of an embedding function is a map $e : V \to E$ that assigns each unit in the vocabulary a numerical representation.

A trivial choice for an embedding function is to choose $E = \{0, 1\}^n$ and assign each element $v_i \in V$ the i-th unit vector in E. This requires one computing element, one integer, for each element v_i and is called a *local representation* (Hinton et al., 1990). In contrast, *distributed representations* use multiple computing elements to represent v_i. An example is to choose E as a d-dimensional Euclidean space with $d \ll n$.

We give an example for a local and distributed representation. Consider a vocabulary containing n elements $V = \{$"go", "went", "explain", "explained", "explanation", $\ldots\}$. A local representation is to assign $e($"go"$) = (1, 0, \ldots)$, $e($"went"$) = (0, 1, \ldots)$, etc. An alternative distributed representation is to consider d dimensional vectors where d is the number of distinct characters across all elements in V. Subsequently we assign $e(v_i)_j = \mathbb{1}\{c_j \in v_i\}$ where $(c_1, c_2, \ldots, c_d)$

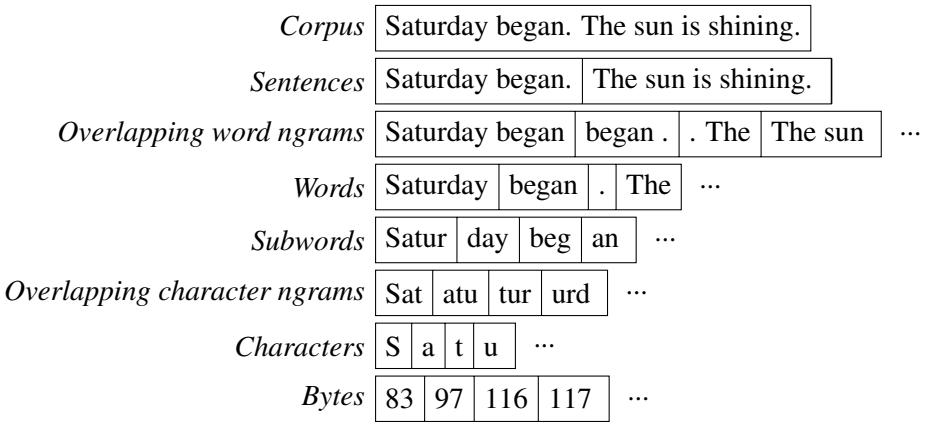


Figure 1.2 – *Common ways to split text data into units. The byte representation depends on the choice of encoding for unicode points (e.g., utf-8).*

is a tuple of all characters. That is, the j-th component of $e(v_i)$ indicates whether unit v_i contains the character c_j.

Some advantages of distributed representations become clear immediately. For the local representation, all vectors are orthogonal and it is impossible to introduce a meaningful similarity metric. For our distributed representation, cosine similarity yields a meaningful metric: the similarity between "explain" and "explained" is higher than between "explain" and "went" as the former share more characters. Adding new elements to distributed representations is easier. For local representation, a new dimension must be added, whereas for distributed representation one can assign a new vector. Local representations are high dimensional and sparse, whereas distributed representations are typically lower dimensional, dense vectors. Empirically, distributed representations have been found to work well with computational models such as logistic regression or neural networks (Hinton et al., 1986; Bengio et al., 2003; Collobert et al., 2011).

However, a major drawback is the difficulty of defining distributed representations. The task of finding a meaningful mapping e is called *representation learning*. Note that it is not clear how to assess whether a mapping is meaningful. Two common ways to judge usefulness are whether the embedding function increases the performance of downstream tasks such as sentiment classification or named entity recognition, or whether similarity measures such as cosine similarity correlate well with semantic or syntactic similarity.

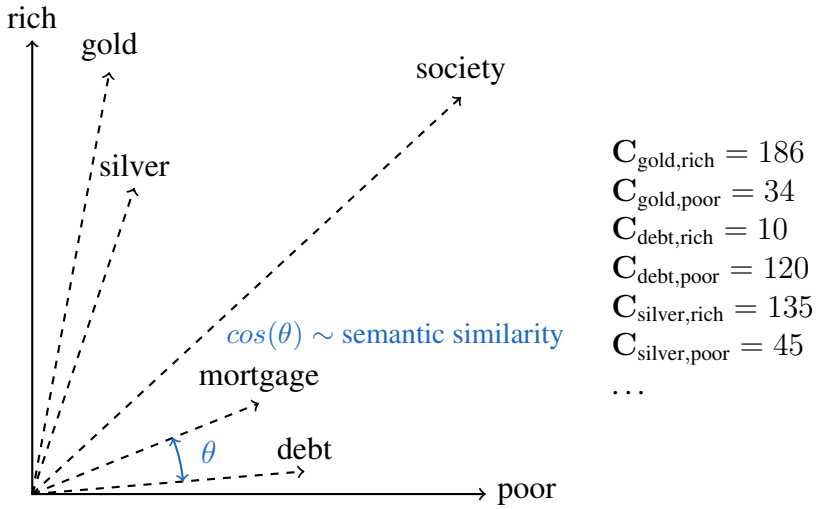


Figure 1.3 – *Example how co-occurrences with other words yield meaningful vector representations. Out of n dimensions we only show the two dimensions corresponding to the units "rich" and "poor".*

1.3 Learning Static Representations

Given a vocabulary V with size n we define a static embedding function as

$$e : V \to \mathbb{R}^d,$$

for some dimensionality $d \in \mathbb{N}$. A central idea to most representation learning algorithms is the hypothesis by Firth (1957) that "you shall know a word by the company it keeps". Loosely speaking, one can infer the meaning of a word by considering all the contexts in which a word appears. Therefore, two words that can occur in similar contexts are likely to have a similar meaning.

It is not immediately clear what it means that two words occur in similar contexts. Given a corpus $U = (u_1, u_2, \ldots, u_t) \in V^t$ that is a sequence of text units we need to introduce a notion of context. A unit u_i occurs in the context of u_j if $|i - j| \leq k$ where k is the size of the context window, i.e., u_i appears in the context of u_j and the units u_i and u_j "co-occur". Note that it is common to segment the corpus U into smaller parts such as sentences. Then, u_i co-occurs with u_j only if u_i is in u_j's context window and both units appear in the same sentence. The *co-occurrence matrix* for the vocabulary V is given by $\mathbf{C} \in \mathbb{N}^{n \times n}$ where $\mathbf{C}_{ij}$ indicates how often the units v_i and v_j co-occur in the corpus U.

1.3.1 Monolingual Representations

A straightforward embedding function is to define $e(v_i) := \mathbf{C}_i$. Figure 1.3 shows an example. That is, the embedding for the unit v_i is the n-dimensional vector that indicates how often v_i co-occurs with every other unit. The advantage of this function is that the cosine similarity of $e(v_i)$ and $e(v_j)$ is high if v_i and v_j occur in similar contexts. Thus, this representation is in line with Firth's hypothesis. A disadvantage is that the dimensionality of $e(v_i)$ is equal to vocabulary size n. As n can be very large this can lead to expensive computations and deteriorated performance when using these embeddings as input to computational models.

Schütze (1992a) proposed to consider matrix factorization, more specifically the singular value decomposition of $\mathbf{C}$. With increasing usage of neural networks in natural language processing, e.g., Collobert et al. (2011), lower dimensional representations became more prevalent and matrix factorizations of $\mathbf{C}$ became more popular, e.g., Levy and Goldberg (2014); Pennington et al. (2014). Mikolov et al. (2013a) presented an alternative way to compute embeddings with shallow neural networks that is widely used now. In the following, we present two popular embedding algorithms, that are relevant to this work, in more detail.

Skip-gram with Negative Sampling

Mikolov et al. (2013a) introduced two methods to estimate word vectors. One method is skip-gram with negative sampling. The underlying idea is to predict, given a unit u_i, whether another unit u_j is likely to appear in the context window of u_i. The computational model is as follows. Consider two matrices $\mathbf{E}, \mathbf{F} \in \mathbb{R}^{n \times d}$ where $\mathbf{E}_{u_i}$ is the word embedding of u_i and $\mathbf{F}_{u_i}$ its context embedding. Further, denote as $c(u_i) \subset V$ the set of tokens that are in the context window of u_i and let $c_n(u_i) \subset V$ be a set of m negative samples, i.e., a set of randomly sampled words that do not occur in the context of u_i. Skip-gram with negative sampling minimizes the following objective

$$\mathcal{L}(\mathbf{E}, \mathbf{F}) = -\sum_{i=1}^{n} \sum_{w \in c(u_i)} \log\left(\sigma(\mathbf{E}_{u_i}^{\intercal} \mathbf{F}_w)\right) - \sum_{w \in c_n(u_i)} \log\left(\sigma(-\mathbf{E}_{u_i}^{\intercal} \mathbf{F}_w)\right), \quad (1.1)$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$. Intuitively, the dot product $\mathbf{E}_{u_i}^{\intercal} \mathbf{F}_w$ should be large when w occurs in the context of u_i and small else. In the actual implementation additional tricks such as down-sampling frequent words or ignoring words that only occur a few times are applied. Eq. 1.1 can then be optimized using stochastic gradient descent. The embedding function is then $e(v) := \mathbf{E}_v$ and the matrix $\mathbf{F}$ is usually ignored. Mikolov et al. (2013a) provided an efficient implementation, and its name, *word2vec*, is sometimes used as synonym for skip-gram with negative sampling.

Query	Nearest Neighbors	
	Skip-gram	Skip-gram with subword information
build	develop, construct, create, rebuild	re-build, rebuild, develop, built
wrote	writes, penned, authored, co-wrote	cowrote, handwrote, hand-wrote, wrote
dish	dishes, cuisine, dessert, recipe	dishes, side-dish, hotdish, one-dish
happy	happier, unhappy, glad, contented	happpy, happy-, happy-happy, unhappy

Table 1.1 – *Four nearest neighbors for a selection of queries obtained from embeddings computed on the English Wikipedia. Clearly, incorporating subword information increases similarity of units that consist of similar ngrams.*

Incorporating Subword Information

A disadvantage of skip-gram is that it only considers the context of each unit and ignores its internal structure. For example the units *building* and *buildings* have separate vectors that are learned independently of each other. Thus, Bojanowski et al. (2017) propose a learning algorithm that incorporates subword information. They provide an implementation of the algorithm in the library *fastText*, which is why fastText became a synonym for this algorithm. Let C_k be the set of all possible combinations of k characters and $\mathcal{G}_k : V \rightarrow \mathcal{P}(C_k)$ be the ngram function that assigns a unit the set of all k-grams that are contained in the unit. For example, $\mathcal{G}_2(day) = \{<d, da, ay, y>\}$. Hyperparameters are the minimum and maximum length of ngrams to consider, denoted by $\underline{k}$ and $\overline{k}$. An additional matrix $\mathbf{G}^{l \times d}$ is introduced, where l is the number of all $\underline{k}$- to $\overline{k}$-grams. Finally, subword information is incorporated in Eq. 1.1 by replacing $\mathbf{E}_{u_i}$ with

$$\mathbf{E}_{u_i} + \sum_{k=\underline{k}}^{\overline{k}} \sum_{g \in \mathcal{G}_k(u_i)} \mathbf{G}_g.$$

That is instead of simply considering a vector for each unit, the representation of a unit is its vector plus the sum (sometimes the average) of the vectors of all its ngrams. Table 1.1 shows the effect of incorporating subword information.

Matrix Factorizations

At first sight, skip-gram does not seem to be related to matrix factorizations of C. Levy and Goldberg (2014) investigated whether skip-gram can be interpreted as matrix factorization of co-occurrences. They identified a relation between skip-gram and the factorization of a shifted pointwise mutual information matrix. That is, the embedding and context matrices $\mathbf{E}$ and $\mathbf{F}$ are a decomposition of a matrix

$\mathbf{M} \in \mathbb{R}^{n \times n}$. More specifically,

$$\mathbf{EF}^\intercal = \mathbf{M}, \quad \text{with} \quad \mathbf{M}_{ij} = log\left(\frac{\mathbf{C}_{ij}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{C}_{ij}}{\sum_{l=1}^{n}\mathbf{C}_{il}\sum_{l=1}^{n}\mathbf{C}_{lj}}\right) - log(m).$$

m is the number of negative samples that is sampled for c_n. While this finding tries to unify matrix factorization approaches with the approach to learn embeddings with shallow neural networks, the statement only holds in a restricted setting, e.g., only for high dimensional embeddings (Arora et al., 2016). In experiments Levy and Goldberg (2014) found, as expected, that skip-gram performs better empirically. One method that combines global matrix factorizations with the local context window approach of skip-gram is *GloVe* (Pennington et al., 2014).

1.3.2 Multilingual Representations

Having described two monolingual representation learning algorithms, we now turn to creating multilingual static embedding spaces. Consider two languages e, f with vocabularies $V^{(e)}$, $V^{(f)}$ of size $n^{(e)}$, $n^{(f)}$ and static embedding spaces $\mathbf{E}^{(e)}$, $\mathbf{E}^{(f)}$ with dimensions $d^{(e)}$, $d^{(f)}$. For the sake of simplicity assume $d^{(e)} = d^{(f)} =: d$.

As with monolingual embedding spaces, it is unclear which properties a multilingual embedding space should exhibit. Intuitively, high quality spaces should enable model transfer, i.e., a computational model trained on embeddings $\mathbf{E}^{(e)}$ should be able to process $\mathbf{E}^{(f)}$ without significant performance decrease, or some measure of similarity like cosine similarity should correlate well with semantic similarity. When both embedding spaces, say in English and German, are learned individually there is no relation between the embeddings. That is $\mathbf{E}^{(e)}_{walk}$ and $\mathbf{E}^{(f)}_{gehen}$ are two separate embeddings with a random cosine similarity. In a multilingual space cos-sim$(\mathbf{E}^{(e)}_{walk}, \mathbf{E}^{(f)}_{gehen})$ should be close to 1 as both units have a similar semantic meaning.

Mapping Approaches

One approach to create multilingual embedding spaces is to learn representations $\mathbf{E}^{(e)}$, $\mathbf{E}^{(f)}$ for each language separately, and subsequently learn some mapping $w : \mathbb{R}^d \to \mathbb{R}^d$ such that the mapped space $\bar{\mathbf{E}}^{(f)}$ with $\bar{\mathbf{E}}^{(f)}_i := w(\mathbf{E}^{(f)}_i)$ and $\mathbf{E}^{(e)}$ are a multilingual space. Figure 1.4 shows this process conceptually. The underlying assumption is that the monolingual spaces have a similar structure. Research has shown that this assumption might only hold for similar languages when the embeddings spaces are trained on comparable training data (Vulić et al., 2020). Assume that we have access to a bilingual dictionary

$$\mathcal{D} := \{(u^{(e)}_1, u^{(f)}_1), (u^{(e)}_2, u^{(f)}_2), \ldots, (u^{(e)}_m, u^{(f)}_m)\},$$

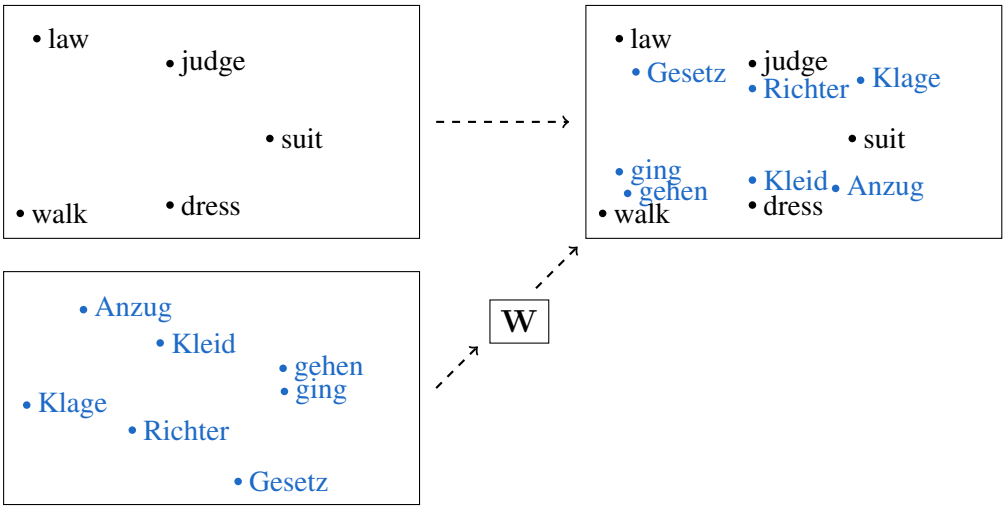


Figure 1.4 – *Mapping a German monolingual embedding space into English embeddings using a rotation matrix* $\mathbf{W}$.

that consists of tuples of translation pairs of units across both languages. Consider the modified embedding matrices $\tilde{\mathbf{E}}^{(e)}$, $\tilde{\mathbf{E}}^{(f)}$ that consist only of embeddings from the dictionary $\mathcal{D}$. A natural approach of creating a multilingual embedding space is to parameterize the function $w(\mathbf{x}) = \mathbf{W}^\intercal\mathbf{x}$ with $\mathbf{W} \in \mathbb{R}^{d\times d}$, and to minimize the objective function

$$\mathcal{L}(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{m}\left\|\tilde{\mathbf{E}}_i^{(e)}\mathbf{W} - \tilde{\mathbf{E}}_i^{(f)}\right\|^2 = \frac{1}{2}\left\|\tilde{\mathbf{E}}^{(e)}\mathbf{W} - \tilde{\mathbf{E}}^{(f)}\right\|_F^2, \tag{1.2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This is an unconstrained quadratic minimization problem, sometimes referred to as the *Procrustes Problem*, with gradient $\nabla\mathcal{L}(\mathbf{W}) = (\tilde{\mathbf{E}}^{(e)})^\intercal(\tilde{\mathbf{E}}^{(e)}\mathbf{W} - \tilde{\mathbf{E}}^{(f)})$. The closed form solution for the unique global optimum is obtained by setting the gradient to $\mathbf{0}$, which yields

$$\mathbf{W}^* = \left((\tilde{\mathbf{E}}^{(e)})^\intercal\tilde{\mathbf{E}}^{(e)}\right)^{-1}(\tilde{\mathbf{E}}^{(e)})^\intercal\tilde{\mathbf{E}}^{(f)}.$$

Mikolov et al. (2013b) investigate mapped multilingual spaces and found them to perform well for word translation. An extension is to constrain the matrix $\mathbf{W}$ to be orthonormal, i.e., $\mathbf{W}^\intercal\mathbf{W} = \mathbf{I}_d$, as introduced in (Xing et al., 2015). This is desirable as an orthonormal transformation does not modify the structure of an embedding space in the sense that it is a norm-preserving transformation: the commonly used euclidean distance and the cosine distance are not affected by the transformation. With this constraint, Eq. 1.2 becomes the *Orthogonal Procrustes Problem* (Schönemann, 1966). A solution is given by computing the singular

value decomposition of the matrix $(\tilde{\mathbf{E}}^{(f)})^\intercal \tilde{\mathbf{E}}^{(e)}$ that is $(\tilde{\mathbf{E}}^{(f)})^\intercal \tilde{\mathbf{E}}^{(e)} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\intercal$. The transformation is then given by $\mathbf{W}^* = \mathbf{V}\mathbf{U}^\intercal$.

While these approaches are simple and effective they require a dictionary $\mathcal{D}$. This is not always easy to obtain and sometimes not even clear how to create (e.g., a dictionary of ngrams). Thus, unsupervised embedding alignment is considered. While unsupervised manifold alignment has been researched for a while in machine learning and computer vision (Wang and Mahadevan, 2009; Cui et al., 2014) its application to word embedding spaces gained popularity with Lample et al. (2018). They propose a method using generative adversarial learning as proposed by Goodfellow et al. (2014). To this end consider some computational model, e.g., a neural network, $f_{\theta_D} : \mathbb{R}^d \rightarrow (0, 1)$ that takes an embedding vector as input. They then consider two objective functions

$$\mathcal{L}_D(\theta_D|\mathbf{W}) = -\frac{1}{n^{(f)}} \sum_{i=1}^{n^{(f)}} \log\left(f_{\theta_D}(\mathbf{W}\mathbf{E}_i^{(f)})\right) - \frac{1}{n^{(e)}} \sum_{i=1}^{n^{(e)}} \log\left(1 - f_{\theta_D}(\mathbf{E}_i^{(e)})\right)$$

$$\mathcal{L}_G(\mathbf{W}|\theta_D) = -\frac{1}{n^{(f)}} \sum_{i=1}^{n^{(f)}} \log\left(1 - f_{\theta_D}(\mathbf{W}\mathbf{E}_i^{(f)})\right) - \frac{1}{n^{(e)}} \sum_{i=1}^{n^{(e)}} \log\left(f_{\theta_D}(\mathbf{E}_i^{(e)})\right).$$

The first loss is the loss of a so called *discriminator*. The discriminator tries to detect whether a vector comes from language f or e. Minimizing this loss means that the model tries to set $f_{\theta_D}(\mathbf{x})$ close to one if the vector is a transformed vector from language f and close to zero if it is a vector from language e. The *generator*, which is simply the function $f_{\mathbf{W}}(\mathbf{x}) = \mathbf{W}\mathbf{x}$, tries to fool the discriminator and has the opposite objective. Both loss functions are optimized alternatingly with gradient descent. For details on the optimization algorithm see (Lample et al., 2018). The resulting multilingual embedding space have been observed to be of low quality. Thus, there is usually a refinement procedure that identifies well-aligned units and uses them as a noisy dictionary $\tilde{\mathcal{D}}$. To get the final transformation $\mathbf{W}^*$ the Procrustes problem is then solved with the noisy dictionary $\tilde{\mathcal{D}}$.

The generator-discriminator setup has been found to be unstable and successful optimization fails frequently (Artetxe et al., 2018). Thus, Artetxe et al. (2018) proposed a fully unsupervised method in their framework *VecMap*. They propose a pipeline of optimization problems and heuristics that results in more robust performance.

Joint Learning

Mapping approaches assume that the embedding spaces $\mathbf{E}^{(e)}$, $\mathbf{E}^{(f)}$ have been learned independently of each other and are subsequently mapped into a common space. Another line of work aims at modifying embedding learning algorithms

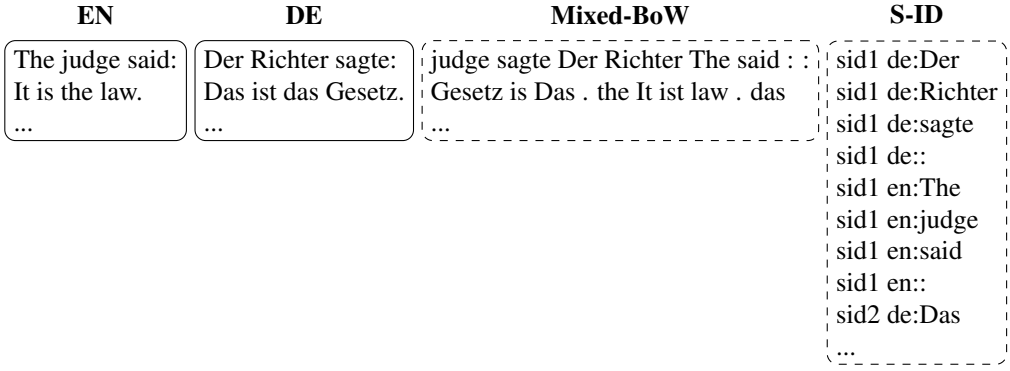


Figure 1.5 – *Transforming parallel corpora for joint learning. Shown are two sentences in a parallel corpus. "Mixed" simply considers the sentences as bags of words. "S-ID" in addition transforms the corpus into tuples of sentence IDs and tokens prefixed with a language identifier. Standard learners such as skip-gram can then be trained on the transformed corpora.*

such that the resulting embeddings $\mathbf{E}^{(e)}$, $\mathbf{E}^{(f)}$ are already multilingual. All of the described methods do also apply to a multilingual and not only a bilingual setting.

First, assume that we have access to a sentence-parallel corpus in two languages. That is we have two corpora in two languages consisting of m sentences $U^{(e)} = (s_1^{(e)}, s_2^{(e)}, \ldots, s_m^{(e)})$, $U^{(f)} = (s_1^{(f)}, s_2^{(f)}, \ldots, s_m^{(f)})$ where $s_i^{(e)} = (u_1^{(e)}, \ldots, u_{t_i^{(e)}}^{(e)})$ and $s_i^{(f)} = (u_1^{(f)}, \ldots, u_{t_i^{(f)}}^{(f)})$ are translations that consist of $t_i^{(e)}$ and $t_i^{(f)}$ units. Two examples of a sentence parallel corpus are the Proceedings of the European Parliament (Koehn, 2005) and the Parallel Bible Corpus (Mayer and Cysouw, 2014). Both are multi-parallel in the sense that translations of the same sentence exist in multiple languages, in the case of the Bible in thousands of languages.

A simple approach of joint learning proposed by Vulić and Moens (2015) is to create *pseudo-bilingual* sentences, a method that we call *Mixed-BoW*. That is, the sentences $s_i^{(e)}$ and $s_i^{(f)}$ are concatenated and randomly shuffled. For example, with two sentences *Du sprichst* and *You are speaking* a sentence like *sprichst You speaking are Du* might be created. Subsequently, a learning algorithm like skip-gram is trained on such a corpus. The underlying intuition is that whole sentences are treated as context for learning word vectors. By creating pseudo-bilingual sentences the tokens *sprichst* and *speaking* occur in similar contexts and thus should result in similar vector representations.

Le and Mikolov (2014) and Dai et al. (2015) found that dense learned representations cannot only represent words but also larger units such as paragraphs. They called this *paragraph vectors*. The underlying idea is to have a classifier

that is able to classify which words are likely to appear in a given paragraph vector. Levy et al. (2017) used the idea of paragraph vectors to propose an algorithm based on sentence IDs which we call *S-ID*. The intuition of S-ID is similar to pseudo-bilingual sentences: units across languages that occur in similar sentences in a parallel corpus are likely to have a similar meaning. Instead of using the individual words as context, S-ID introduces one vector that represents whole sentences, similar to paragraph vectors. More specifically a modified corpus as depicted in Figure 1.5 is created and subsequently, skip-gram is trained on such a corpus. Levy et al. (2017) found that this is a strong baseline for learning multilingual embeddings.

Other approaches for joint learning modify the objective function of skip-gram to incorporate crosslingual signals such as a dictionary during training, e.g., Multiskip or Multicluster by Ammar et al. (2016).

1.4 Learning Contextualized Representations

Static representations are a mapping $e : V \rightarrow \mathbb{R}^d$, thus each unit in the vocabulary is assigned a single vector. More specifically the unit *break* has the same representation, whether it refers to *taking a break* or *to break the silence*. It is not immediately clear whether conflating multiple meanings of an ambiguous unit in a single vector is harmful.

Contextualized representations take the context, in which a unit appears, explicitly into account. We define a contextualized embedding function as

$$e : V^{t_{max}} \rightarrow \mathbb{R}^{t_{max} \times d},$$

where t_{max} is the maximum number of units that the function can process at once. Thus, the contextualized embedding of the unit u_i in a sentence $(u_1, \ldots, u_i, \ldots u_t)$ depends on all other units in the sentence, and the representations for the unit *break* in the above examples differ.

1.4.1 Monolingual Representations

In this section, we present several methods to obtain contextualized monolingual representations.

Pretrained Language Models

Among the first approaches to learn contextualized embeddings are (Peters et al., 2017; McCann et al., 2017). The central idea is to learn a neural language model and use the hidden states of the model as contextualized embeddings.

A language model models the probability that a sequence of units occurs, i.e., it models the probability $P(u_1, u_2, \ldots, u_t)$. Usually the sequential nature of text is taken into account and common factorizations of the probability are obtained by applying the chain rule of probabilities. A forward and backward language model is given by

$$P(u_0, u_1, u_2, \ldots, u_t, u_{t+1}) = P(u_0) \prod_{i=0}^{t} P(u_{i+1}|u_0, \ldots, u_i)$$

$$P(u_0, u_1, u_2, \ldots, u_t, u_{t+1}) = P(u_{t+1}) \prod_{i=0}^{t} P(u_i|u_{t+1}, \ldots, u_{i+1}),$$

where u_0 and u_{t+1} are tokens added that indicate the start and end of the sequence.

A statistical model $P_\theta(u_{i+1}|u_0, \ldots, u_i)$ can then be for example parameterized with a recurrent neural network (RNN). A RNN is a model that is recursively computed as

$$\begin{aligned} \mathbf{h}^{(i)} &= \sigma_h(\mathbf{W}^{(u)}\mathbf{e}^{(i)} + \mathbf{W}^{(h)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(h)}) \\ \mathbf{y}^{(i+1)} &= \sigma_y(\mathbf{W}^{(y)}\mathbf{h}^{(i)} + \mathbf{b}^{(y)}) \end{aligned}$$

for $i = 1, \ldots, t$, where $P_\theta(u_{i+1}|u_0, \ldots, u_i) = \hat{\mathbf{y}}^{(i+1)}_{u_{i+1}}$ is the probability for the unit occurring at u_{i+1},

$$\theta = (\mathbf{W}^{(u)}, \mathbf{W}^{(h)} \in \mathbb{R}^{d\times d}; \mathbf{h}^{(0)}, \mathbf{b}^{(h)} \in \mathbb{R}^d; \mathbf{E}, \mathbf{W}^{(y)} \in \mathbb{R}^{n\times d}; \mathbf{b}^{(y)} \in \mathbb{R}^n)$$

are the parameters of the model, $\mathbf{e}^{(i)}$ is the an embedding vector for the unit u_i, i.e., $\mathbf{e}^{(i)} = \mathbf{E}_{u_i}$, and $\sigma : \mathbb{R}^m \to \mathbb{R}^m$ are activation functions. A common choice is applying tangens hyperbolicus $\sigma_h(\mathbf{x})_k = \tanh(\mathbf{x}_k)$ component-wise and the softmax function $\sigma_y(\mathbf{x})_k = e^{\mathbf{x}_k} / \sum_{i=1}^{n} e^{\mathbf{x}_i}$. Usually, as objective function the negative cross entropy between the $\hat{\mathbf{y}}^{(i+1)}$ and the observed units is used. That is

$$\mathcal{L}(\theta, U) = -\frac{1}{m} \sum_{k=1}^{m} \sum_{i=1}^{t_k} \log(\hat{\mathbf{y}}^{(i)}_{u_i}),$$

where $U = (s_1, \ldots, s_m)$ is a corpus with m sentences with $s_k = (u_1, \ldots, u_{t_k})$. The objective function $\mathcal{L}$ is then minimized using for example stochastic gradient descent.

In practice, such plain RNNs are difficult to train because of misbehaved gradients (Bengio et al., 1994). Variants like Long Short-Term Memory (*LSTM*) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (*GRU*) (Cho et al., 2014b) are more popular and have been shown to perform better for language

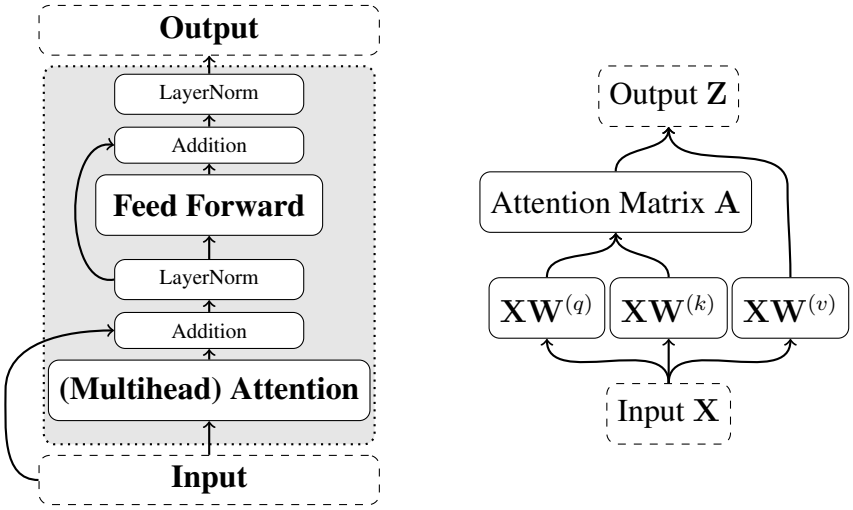


Figure 1.6 – *Left: Schema of a transformer encoder block (grey block) that can be repeated for l layers. Right: schematic description of the Self-Attention computation.*

modeling, e.g., (Sundermeyer et al., 2012). Peters et al. (2017) proposed to train a language model that consists of a forward and backward LSTM. They then considered the hidden states of the neural network as embeddings and used them together with static embeddings as input to models that solved a downstream task, such as named entity recognition. This approach yielded state of the art performance and was further developed: Peters et al. (2018) introduced deep contextualized embeddings called *Embeddings from Language Models* (*ELMo*) and outperformed the state of the art performance across many tasks. In ELMo, bidirectional LSTMs are trained with the task of language modeling. One novelty is to let the downstream task model learn a weighted combination of hidden representations across different layers in ELMo.

A big advantage is that language models do not need any manually labeled data and can thus be (pre-)trained on large amounts of text data obtained for example from the internet. Subsequently, these models can be used for downstream tasks. This motivates the name *Pretrained Language Models* (PLMs).

Transformer Models

Recurrent neural networks have a sequential nature by definition. However, propagating information across long time spans turned empirically out to be difficult, e.g., analyzed by Cho et al. (2014a) for machine translation. Extensions such as *attention* have been added to overcome this issue (Bahdanau et al., 2015). Vaswani et al. (2017) proposed a machine translation system that uses attention only: it ex-

hibits superior performance compared to recurrent neural networks. More specifically, they propose to stack *Transformer Encoder Blocks* as shown in Figure 1.6. One such block is a function $t_\theta : \mathbb{R}^{t\times d} \to \mathbb{R}^{t\times d}$ with $t_\theta(\mathbf{X}) =: \mathbf{Z}$ that is computed as follows:

$$\begin{aligned}
\mathbf{A} &= \sqrt{\frac{1}{d_h}}\mathbf{X}\mathbf{W}^{(q)}(\mathbf{X}\mathbf{W}^{(k)})^\intercal \\
\mathbf{M} &= \text{SoftMax}(\mathbf{A})\mathbf{X}\mathbf{W}^{(v)} \\
\mathbf{O} &= \text{LayerNorm}_1(\mathbf{M} + \mathbf{X}) \\
\mathbf{F} &= \text{ReLU}(\mathbf{O}\mathbf{W}^{(f_1)} + \mathbf{b}^{(f_1)})\mathbf{W}^{(f_2)} + \mathbf{b}^{(f_2)} \\
\mathbf{Z} &= \text{LayerNorm}_2(\mathbf{O} + \mathbf{F}),
\end{aligned}$$

where $\text{SoftMax}(\mathbf{A})_{ij} = e^{\mathbf{A}_{ij}} / \sum_{k=1}^{t} e^{\mathbf{A}_{ik}}$ is the softmax function applied row-wise, $\text{LayerNorm}(\mathbf{X})_i = \mathbf{g} \odot (\mathbf{X}_i - \mu(\mathbf{X}_i))/\sigma(\mathbf{X}_i) + \mathbf{b}$ is layer normalization with $\mu(\mathbf{x}), \sigma(\mathbf{x})$ returning the mean and standard deviation of a vector, and $\text{ReLU}(\mathbf{X}) = \max(0, \mathbf{X})$ is the maximum operator applied component-wise. In this notation we assume broadcasting as implemented in NumPy (Harris et al., 2020) when for example adding a vector to a matrix. The parameters of such a block are

$$\begin{aligned}
\theta = (&\mathbf{W}^{(q)}, \mathbf{W}^{(k)}, \mathbf{W}^{(v)} \in \mathbb{R}^{d\times d}; \mathbf{g}^{(1)}, \mathbf{g}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)} \in \mathbb{R}^{d}; \\
&\mathbf{W}^{(f_1)} \in \mathbb{R}^{d\times d_f}; \mathbf{W}^{(f_2)} \in \mathbb{R}^{d_f\times d}; \mathbf{b}^{(f_1)} \in \mathbb{R}^{d_f}; \mathbf{b}^{(f_2)} \in \mathbb{R}^{d}),
\end{aligned}$$

where d is called the hidden dimension, d_f the intermediate dimension, and t is the sequence length. Usually, multiple, say h, attention heads are considered, that is $\mathbf{W}^{(q)}, \mathbf{W}^{(k)}, \mathbf{W}^{(v)} \in \mathbb{R}^{d\times d_h}$ where $d = hd_h$. The matrices $\mathbf{M}^{(h)} \in \mathbb{R}^{t\times d_h}$ are then concatenated along the second dimension to obtain $\mathbf{M}$. We call a *Transformer* a function $T_\theta : \mathbb{R}^{t\times d} \to \mathbb{R}^{t\times d}$ that consists of multiple applications of transformer blocks, i.e., $T_\theta(\mathbf{X}) = t_{\theta^l} \circ t_{\theta^{l-1}} \circ \cdots \circ t_{\theta^1}(\mathbf{X})$. We say that the encoder has l layers.

Consider now how a Transformer is applied to text data, for example the sequence $U = (u_1, u_2, \ldots, u_t)$. First, unit embeddings $\mathbf{U} \in \mathbb{R}^{t\times d}$ are created by a lookup in the embedding matrix $\mathbf{E} \in \mathbb{R}^{n\times d}$ where n is the vocabulary size. When analyzing a Transformer closely one observes that it is invariant with respect to reorderings of the input. To counteract this effect, positional encodings are added. That is, a matrix of position embeddings $\mathbf{P} \in \mathbb{R}^{t\times d}$ is created. The final input to T_θ is then $\mathbf{U} + \mathbf{P}$. Both $\mathbf{E}$ and $\mathbf{P}$ are learnable parameters of the model. Sometimes additional embeddings such as segment or language embeddings are added to the input, as well.

Input	Top Predictions
Today is [MASK] weather.	good, the, bad, fair, dry
No, this is not [MASK].	true, right, it, happening, possible
The egg fell on the floor and [MASK].	shattered, exploded, disappeared, fell, broke
The capital of France is [MASK].	Paris, Lyon, Toulouse, Lille
Yesterday, he [MASK] me the money.	gave, handed, left, offered

Table 1.2 – *Top predictions for some masked sentences using the pretrained BERT model "BERT-base-cased" (Devlin et al., 2019).*

Bidirectional Encoder Representations from Transformers

PLMs and the Transformer culminated in the development of *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin et al., 2019). BERT is a Transformer model that is pretrained on a variant of language modeling. The main innovation is that BERT is not a unidirectional language model, but rather bidirectional. Instead of only having access to the right or left context of a word, BERT has access to context words on both sides simultaneously. To this end they propose to use *masked language modeling* (*MLM*). Consider again a corpus $U = (u_1, \ldots, u_t)$. Further, consider a sequence of independent and identically distributed (iid) Bernoulli random variables $(B_i)_{i=1,\ldots,t}$, a sequence of iid random variables $(R_i)_{i=1,\ldots,t}$ with a uniform distribution on $[0, 1]$ and a sequence of iid random variables $(W_i)_{i=1,\ldots,t}$ that are uniformly distributed on $\{1, 2, \ldots, n\}$. A modified version of the corpus U denoted by U' is then created by setting

$$u'_i = \begin{cases} u_i & \text{if } B_i = 0 \\ \text{[MASK]} & \text{if } B_i = 1 \wedge R_i \leq \rho_1 \\ v_{W_i} & \text{if } B_i = 1 \wedge \rho_1 < R_i \leq \rho_2 \\ u_i & \text{if } B_i = 1 \wedge \rho_2 < R_i, \end{cases}$$

where v_{W_i} is essentially a randomly sampled unit from the vocabulary and [MASK] is a special token. A typical value is $P(B = 1) = 0.15$ and $\rho_1 = 0.8, \rho_2 = 0.9$. The modified corpus is then used as input to a Transformer. When $\mathbf{X}$ is a modified input to the Transformer model, it can be used to predict the original input. To this end the output $T_\theta(\mathbf{X})$ is once more transformed to obtain $S(\mathbf{X}) = \text{LayerNorm}(\sigma(T_\theta(\mathbf{X})\mathbf{W}^{(s)} + \mathbf{b}^{(s)}))$ where $\mathbf{W}^{(s)} \in \mathbb{R}^{d\times d}$, $\mathbf{b}^{(s)} \in \mathbb{R}^d$ are parameters and σ is some activation function applied component-wise. In order to get token predictions the token embeddings from the first layer are reused. Thus, the prediction scores are obtained by $P(\mathbf{X}) = \text{SoftMax}(S(\mathbf{X})\mathbf{E}^\intercal + \mathbf{b}_p)$ where $\mathbf{b}_p \in \mathbb{R}^n$ is a token-specific bias. For each position i, $P(\mathbf{X})_i$ can be interpreted as a distribution over vocabulary units. That is, $\hat{p}_{u_i} = P(\mathbf{X})_{i,u_i}$ indicates the probability that the

model predicts the correct unit u_i at position i. The objective function is

$$\mathcal{L}(\theta, \psi, U, U') = -\frac{1}{t}\sum_{i=1}^{t} \log(\hat{p}_{u_i})\mathbb{1}_{B_i}, \tag{1.3}$$

where ψ are all parameters from the prediction head as described above. Note that in practice the input is again split into shorter sequences such as sentences. BERT is then trained by minimizing the objective function in Eq. 1.3. Table 1.2 shows some predictions by a pretrained model.

There are additional technical aspects. BERT uses a wordpiece tokenizer (Schuster and Nakajima, 2012) to split text into subword units. This is advantageous as it reduces the memory requirements of the embedding matrix $\mathbf{E}$ while enabling the model to cover a large variety of text sequences. Further, the Adam optimizer (Kingma and Ba, 2015) is used and regularization methods such as dropout (Srivastava et al., 2014) are integrated into the model. The original BERT model has a second loss term that deals with the task of *next sentence prediction*. However, it has been found that this term does not contribute to the performance of the model (Liu et al., 2019). Thus, this term is mostly omitted and we do not describe it here.

The principle of using BERT for downstream tasks is then similar to standard static embeddings. For sequence labeling tasks the output probabilities per token are usually obtained by adding a feed forward layer, called a prediction head, on top of the pretrained transformer, that is $P(\mathbf{X}) = \text{SoftMax}(T(\mathbf{X})\mathbf{W}_p + \mathbf{b}_p)$, where $\mathbf{W}_p \in \mathbb{R}^{d \times n_l}$ and $\mathbf{b}_p \in \mathbb{R}^{n_l}$ for n_l possible labels in the task. All parameters, from this new task-specific prediction head, which we denote again by ψ and the parameters θ from the pretrained Transformer model, are then jointly trained by optimizing a loss function that is suitable for the downstream task (e.g., categorical cross-entropy). This process is called *fine-tuning*. For sequence classification typically only the first output vector of a sequence, the one corresponding to the start-of-sequence token is used.

Alternative Pretrained Language Models

ELMo and BERT initiated the development of a huge variety of pretrained language models. Since then pretrained models with different sizes, model architectures, and trained on different data have been released. We mention some of the most important models here. The Transformer decoder based GPT-family (Radford et al., 2019; Brown et al., 2020) is well suited for text generation. They have been shown to solve tasks with very few training examples. With ALBERT, Lan et al. (2020) propose weight sharing across layers, factorizing the embedding lookup matrix, and using a more challenging alternative to the next sequence prediction task. A different approach is pursued by ELECTRA (Clark et al., 2020b),

a model that consists of a generator and discriminator part. The generator performs the mask language modeling task whereas the discriminator needs to predict which tokens are from the original text data. Other interesting models are BART (Lewis et al., 2020), that is trained by corrupting the input text with multiple transformations, or T5 (Raffel et al., 2020) focused on transfer learning. Throughout this work, we mostly focus on BERT.

1.4.2 Multilingual Representations

With BERT there is an effective method for learning high quality contextualized representations. We are now interested in *multilingual* contextualized representations.

Joint Training

A multilingual BERT (*mBERT*) version has been presented in the context of (Devlin et al., 2019). The underlying idea is simple: consider Wikipedia data across multiple languages (in the case of mBERT, 104 languages) denoted by $U_{l_1}, U_{l_2}, \ldots$. A shared vocabulary across these corpora is learned, that is, individual tokens can appear in multiple languages (e.g., *end* can appear both in the English word *ending* and the German word *enden*). Subsequently, the data from all Wikipedias is concatenated and shuffled and a standard BERT model is trained. There is no crosslingual supervision in terms of parallel data or a dictionary. Also, the loss function does not exhibit any term that encourages the model to become multilingual. Technical details include for example up- and downsampling of the training corpus to achieve better performance for languages with small Wikipedias. This model has been found to yield good multilingual representations when evaluating them with methods described in Section 1.5.

Alternative Approaches

There is a range of work that tries to improve the multilinguality of mBERT. *Translation language modeling* proposed by Conneau and Lample (2019) uses parallel sentences. The input to a Transformer model can look like "The sun shines. - Die [MASK] scheint." The model can now use the English context to predict the unit "Sonne" in the German sentence. Intuitively, this should increase the multilinguality of the model. Related to this, Cao et al. (2020) introduce a regularization term that encourages the model to get similar representations for words that are aligned in a parallel corpus. In the case of the example, the representations of "sun" and "Sonne" are encouraged to have a high cosine similarity. Conneau et al. (2020a) do crosslingual learning at scale: they use more data, a larger shared

vocabulary, and omit the next sequence prediction term. Similar to monolingual mapping approaches Conneau et al. (2020b) showed that monolingual contextualized embeddings can be mapped into a common multilingual model using linear transformations. Throughout this thesis, we mostly focus on mBERT.

1.5 Evaluating Multilingual Representations

So far we have discussed how to obtain static and contextualized representations both monolingually and multilingually. The actual objectives, however, have been vague: (a) how to evaluate the quality of representations, (b) what is a language, (c) and when are representations multilingual?

(a) In the monolingual case, the quality of static embeddings is typically assessed using *intrinsic* and *extrinsic* evaluation. Intrinsic evaluation assesses properties of the embeddings using tasks like word similarity, e.g., (Bruni et al., 2014; Hill et al., 2015; Gerz et al., 2016), word analogy, e.g., (Mikolov et al., 2013c; Gladkova et al., 2016) or correlation with linguistic features, e.g., QVEC by Tsvetkov et al. (2015). With extrinsic evaluation, embeddings are used as input to a model that solves a downstream task, like part-of-speech tagging or named entity recognition. Monolingual contextualized embeddings can be evaluated using perplexity in language modeling. More commonly used, however, is a range of extrinsic tasks for which the model is finetuned, e.g., the GLUE or SuperGLUE benchmarks (Wang et al., 2018, 2019) for natural understanding or again tasks like named entity recognition.

(b) The decision whether text is written in different languages is not as clear as it seems at first sight. Edge cases are for example heavily code-mixed text (i.e., a mix of two languages), dialects (e.g., Bavarian), and specialized domain language (e.g., a contract vs. a tweet). For the sake of simplicity, in this thesis, we say that data is from two different languages if it has been assigned different ISO-639-3 codes.

(c) For evaluating the multilingual representations consider the following use cases for which they are most likely useful. (i) **Translation.** If representations of units (e.g., words, phrases or sentences), that are semantically similar, are close to each other across languages, they can be used e.g., for word translation using simple similarity search in a numerical space. Other applications include crosslingual sentence retrieval or word alignment. (ii) **Zero-shot transfer.** Consider multilingual representations, a model that uses these representations and is only trained on English data for a specific downstream task such as named entity recognition. If the multilingual representations now allow the model not only to recognize entities in English but also in other languages we call this *zero-shot transfer* across languages. This is useful in practice as annotated training data for downstream

tasks involves manual labor: annotating data for 100s or 1000s languages is a tedious, expensive, and error-prone process which is why zero-shot transfer is useful for practitioners. (iii) **Low-resource processing.** Most text data is available in English. However, there is a large number of languages for which few data is available, called *low-resource* languages. Training a multilingual model such that downstream task performance for low-resource languages is improved is a central objective of multilingual NLP. (iv) **Unified modeling.** Last, even in the case of abundant data in all languages, having a single statistical model that can process multiple languages is desirable as it is easier to maintain than having hundreds of monolingual models.

1.5.1 Intrinsic Evaluation

Throughout this section, we assume we have access to static and contextualized multilingual embeddings.

Word Translation

Consider a bilingual dictionary $\mathcal{D} = \{(v_1^{(e)}, v_1^{(f)}), (v_2^{(e)}, v_2^{(f)}), \ldots, (v_m^{(e)}, v_m^{(f)})\}$ and the vocabularies $V^{(e)}, V^{(f)}$ for two languages e, f with size n_e, n_f. Let $\mathbf{E}^{(e)} \in \mathbb{R}^{n_e \times d}$, $\mathbf{E}^{(f)} \in \mathbb{R}^{n_f \times d}$ be embeddings for each unit in the vocabulary. For contextualized embedding models these can be obtained by feeding each unit without any context into the model. A translation for a query $v_i^{(e)}$ is then obtained by considering the nearest neighbor in the embedding space of language f, i.e.,

$$\hat{v}_i^{(f)} = \arg \max_{j=1,\ldots,n_f} \text{cos-sim}(\mathbf{E}^{(e)}_{v_i^{(e)}}, \mathbf{E}^{(f)}_{v_j^{(f)}}).$$

Instead of cosine similarity other similarity measures such as cross-domain similarity local scaling (Lample et al., 2018) have been found to yield better results. The evaluation metric is *hits at one*, which we define as

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{\hat{v}_i^{(f)} = v_i^{(f)}\}.$$

Analogously when not only retrieving the nearest neighbor but the list of k nearest neighbors one can compute *hits at k*. Word translation datasets can for example be found in (Dinu and Baroni, 2015) or (Lample et al., 2018).

Sentence Retrieval

Similar to word translation, one can evaluate representations using sentence retrieval. Instead of a dictionary one now considers parallel sentences

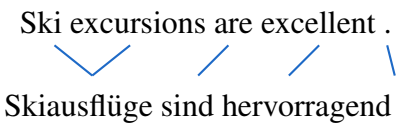


Figure 1.7 – *Example of a word alignment.*

$\mathcal{U} = \{(s_1^{(e)}, s_1^{(f)}), (s_2^{(e)}, s_2^{(f)}), \ldots, (s_m^{(e)}, s_m^{(f)})\}$. Let $\mathbf{E}^{(s_i^{(e)})}$ be the embeddings of sentence $s_i^{(e)} = (u_1, \ldots, u_{t_i})$ that consists of t_i units. One can then obtain a sentence representation by simply averaging across unit representation, i.e.,

$$\mathbf{e}_{s_i^{(e)}} = \frac{1}{t_i} \sum_{j=1}^{t_i} \mathbf{E}_j^{(s_i^{(e)})}.$$

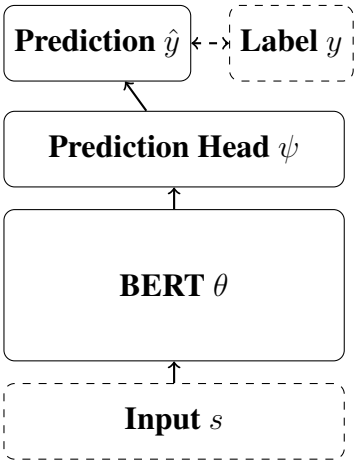
Once sentence representations are obtained the evaluation is identical to word translation. Popular benchmarks are from the BUCC shared task (Zweigenbaum et al., 2017) or the Tatoeba test set by Artetxe and Schwenk (2019).

Word Alignment

Word alignment is a task where translations of units in two parallel sentences should be identified, see Figure 1.7 for an example. Consider the same parallel corpus as for sentence retrieval. For a parallel sentence pair $s_i^{(e)}, s_i^{(f)}$ a word alignment is a bipartite graph, where the units in $s_i^{(e)}$ and $s_i^{(f)}$ are the nodes of the partitions in the graph denoted by $V_i^{(e)}, V_i^{(f)}$. Typically, there is a set of sure and possible edges, $S_i, P_i \subset V_i^{(e)} \times V_i^{(f)}$ where $S_i \subset P_i$. In an alignment gold standard the edges are manually created. The task is to predict the edges automatically, i.e., create a set of predicted edges A_i. Edge sets without index denote the union of all edges across all sentences, i.e., $S = \bigcup_{i=1}^{m} S_i$. Standard evaluation metrics are then precision, recall, F_1 and alignment error rate (AER) (Och and Ney, 2000) computed by

$$\text{prec} = \frac{|A \cap P|}{|A|}, \quad \text{rec} = \frac{|A \cap S|}{|S|}, \quad F_1 = \frac{2 \text{ prec rec}}{\text{prec} + \text{rec}},$$
$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}.$$

In the simplest case, the predicted alignment edges A_i for a sentence can be obtained by performing nearest neighbor search in an embedding space restricted on the units that occur in the sentences.



Stage	*Updating Parameters*	*Data*
Pretraining	θ	$\mathcal{U}^{(eng)}, \mathcal{U}^{(spa)}, \mathcal{U}^{(zho)}, \ldots$
Finetuning	ψ, θ	$\mathcal{D}^{(eng)}_{\text{train}}$
Evaluation		$\mathcal{D}^{(eng)}_{\text{test}}$
Zero-shot		$\mathcal{D}^{(spa)}_{\text{test}}, \mathcal{D}^{(zho)}_{\text{test}}, \ldots$

Figure 1.8 – *Overview of extrinsic evaluation with zero-shot language transfer. $\mathcal{U}$ is some text data, e.g., Wikipedia in the corresponding languages and $\mathcal{D}$ is labeled training or test data for a specific task such as natural language inference or named entity recognition.*

1.5.2 Extrinsic Evaluation

In principle, any downstream task can be considered for multilingual evaluation as long as task data in multiple languages is available. Some popular benchmarks are: natural language inference (*NLI*), e.g., XNLI by Conneau et al. (2018). NLI is a classification task where an ordered sentence pair s_p, s_h, a premise and a hypothesis, is assigned a label out of $\{$neutral, entailment, contradiction$\}$. The predictions are then evaluated using accuracy. Other tasks are named entity recognition, e.g., the CoNLL shared task 2003 (Tjong Kim Sang and De Meulder, 2003) or Wikiann by Pan et al. (2017), or question answering, e.g., TyDi by Clark et al. (2020a) or XQuAD by Artetxe et al. (2020a). For additional tasks see (Hu et al., 2020); they present *XTREME*, a benchmark consisting of nine task.

Zero-shot Transfer

Consider a setup where training data consisting of text sequences and corresponding labels in one language is available,

$$\mathcal{D}^{(e)}_{\text{train}} = \left((y^{(e)}_1, s^{(e)}_1), \ldots, (y^{(e)}_k, s^{(e)}_k) \right),$$

and test data in multiple languages, $\mathcal{D}^{(e)}_{\text{test}}, \mathcal{D}^{(f)}_{\text{test}}, \mathcal{D}^{(g)}_{\text{test}}, \ldots$. The basic idea now is to model the probability that a sequence s has labels y, i.e., $P(y|s)$, for all languages with a function $p_\phi(s) = c_\psi \circ e_\theta(s)$. Here, $e_\theta(s)$ is a pretrained multilingual embedding function such as multilingual BERT. The parameters θ are learned during the

pretraining stage where unlabeled data $\mathcal{U}^{(e)}, \mathcal{U}^{(f)}, \mathcal{U}^{(g)}, \ldots$ is used. In the *finetuning* stage the parameters $\phi = (\psi, \theta)$ are learned using $\mathcal{D}^{(e)}_{\text{train}}$, i.e., the parameters θ are further finetuned. Once the optimal parameters, denoted by a star, are obtained, the model $p_{\phi^\star}(s)$ is evaluated on the test data for all languages. The last stage is called *zero-shot transfer*. See Figure 1.8 for an overview.

Low-resource Processing

Again, downstream tasks are considered. Consider unlabeled data $\mathcal{U}^{(r)}$ and labeled data $\mathcal{D}^{(r)}_{\text{train}}, \mathcal{D}^{(r)}_{\text{test}}$ in a low-resource language r (i.e., the datasets have few instances). One objective of multilingual representation learning is to use data from another language $\mathcal{U}^{(e)}$ together with $\mathcal{U}^{(r)}$ to learn multilingual representations. When these representations are then used for solving the downstream task, the objective is to increase the performance on $\mathcal{D}^{(r)}_{\text{test}}$.

Unified Modeling

In this setup training and test data is available in all considered languages. The objective is to model several languages together without performance decrease compared to an individual model per language.

1.6 Conclusion

This introductory chapter was meant to describe basic concepts in the field of multilingual representation learning that are relevant to this thesis. We motivated multilingual representations and set mathematical and linguistic foundations. Further, we have presented existing methods on how to learn static and contextualized representations for textual units. Multilinguality can be achieved through joint training or post-hoc mapping, either unsupervised or by using crosslingual signals. Last, we outlined methods to evaluate the degree of multilinguality of such representations. Throughout the next chapters, we analyze and use the presented methods in a series of research papers.

1.6.1 Contributions

In light of our four research questions posed at the beginning we can categorize and summarize our contributions in this thesis as follows.

i) *Properties:* We showed that pretrained multilingual language models exhibit a high similarity across languages on the token level in the middle layers. Further, we created a multilingual dataset to investigate factual knowledge in language models.

ii) *Limits:* We explored the limits of multilingual representations by creating embedding spaces that cover 1259 languages. In addition, we showed that multilinguality research can be feasible in a small and computationally efficient setup.

iii) *Analysis:* We proposed a closed form solution for interpreting distributed representations and identified factors that influence the multilinguality of language models trained without any multilingual supervision.

iv) *Improvements:* We developed concept based embedding learning tailored for learning massively multilingual static embeddings. Further, we proposed a masking scheme based on code-switching that achieves a higher degree of multilinguality in pretrained language models.

1.6.2 Limitations

Naturally, the scope of this work is quite limited and we can only deal with a small part of the research questions that are outlined in Section 1.1.2. Many important aspects in the field of multilingual distributed representations have not been addressed. For example, the definition of *language* has not been questioned. Further,

we pursue a language agnostic approach. Investigating and incorporating linguistic properties and similarities across languages is a promising research approach for future work. Similarly, using non-textual crosslingual signals is an interesting approach that is beyond the scope of this work. Last, one can ask the question whether multilingual models are necessary at all or whether it is a more promising path to machine translate data from all languages into English for automated processing.

1.6.3 Future Work

For each of the four research questions we describe potential future work and related literature that addresses these research directions.

i) The objective of multilingual representation learning can be made more specific. The ultimate goal of multilingual learning is quite broad and specific objectives are often somewhat vague. For example, it is not completely clear whether high similarity of semantically similar units and good zero-shot transfer capabilities describe the same objective. As a consequence the evaluation setup currently used might not be ideal and could be improved (Glavaš et al., 2019; Artetxe et al., 2020b). Further, it could be investigated whether multilingual representations can be separated into language-specific and language-agnostic parts, e.g., along the lines of (Pires et al., 2019; Libovický et al., 2020), and whether this is beneficial or harmful for multilingual applications.

ii) Limits of multilingual embedding spaces can be investigated. For example, it is unclear which units across languages are compatible and can be represented in a single space. Latin characters in English are somewhat similar to Cyrillic characters but quite different from Chinese. Thus, one can hypothesize that English character representations are somewhat incompatible with Chinese character representations. Subword representations in English, however, might be more similar to character representations in Chinese. Loosely related work in that direction investigates isomorphy of embeddings across languages (Vulić et al., 2020).

iii) In order to assess the quality and differences of existing algorithms, new evaluation resources can be created. While there is a recent surge of resources, e.g., (Roy et al., 2020; Clark et al., 2020a; Ponti et al., 2020), and datasets with good language coverage exist for some tasks like part-of-speech tagging and named entity recognition (Nivre et al., 2020; Rahimi et al., 2019), there is room for creating new datasets across a variety of tasks with an increased language coverage and language diversity.

iv) Last, a higher degree of multilinguality can be achieved by using existing crosslingual resources more effectively. That is, to use readily available word dictionaries and parallel sentences rather than focusing on the somewhat artificial scenario of unsupervised multilingual representation learning (Vulić et al., 2019). Similarly, zero-shot transfer is, albeit being interesting from an academic perspective, somewhat artificial in reality as it seems feasible to obtain at least a few annotated training samples per language. Few-shot training, i.e., using a small number of annotated samples for each language generally yields improved performance (Lauscher et al., 2020).

Chapter 2

Embedding Learning Through Multilingual Concept Induction

Embedding Learning Through Multilingual Concept Induction

Philipp Dufter[1], Mengjie Zhao[2], Martin Schmitt[1], Alexander Fraser[1], Hinrich Schütze[1]
[1] Center for Information and Language Processing (CIS) LMU Munich, Germany
[2] École Polytechnique Fédérale de Lausanne, Switzerland
{philipp,martin,fraser}@cis.lmu.de, mengjie.zhao@epfl.ch

Abstract

We present a new method for estimating vector space representations of words: embedding learning by concept induction. We test this method on a highly parallel corpus and learn semantic representations of words in 1259 different languages in a single common space. An extensive experimental evaluation on crosslingual word similarity and sentiment analysis indicates that concept-based multilingual embedding learning performs better than previous approaches.

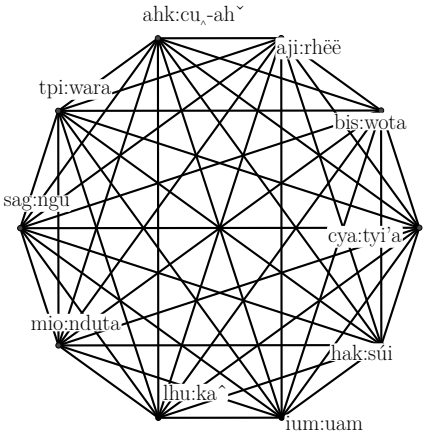


Figure 1: Example of a CLIQUE concept: "water"

1 Introduction

Vector space representations of words are widely used because they improve performance on monolingual tasks. This success has generated interest in multilingual embeddings, shared representation of words across languages (Klementiev et al., 2012). Such embeddings can be beneficial in machine translation in sparse data settings because multilingual embeddings provide meaning representations of source and target in the same space. Similarly, in transfer learning, models trained in one language on multilingual embeddings can be deployed in other languages (Zeman and Resnik, 2008; McDonald et al., 2011; Tsvetkov et al., 2014). Automatically learned embeddings have the added advantage of requiring fewer resources for training (Klementiev et al., 2012; Hermann and Blunsom, 2014b; Guo et al., 2016). Thus, massively multilingual word embeddings (i.e., covering 100s or 1000s of languages) are likely to be important in NLP.

The basic information many embedding learners use is *word-context information*; e.g., the embedding of a word is optimized to predict a representation of its context. We instead learn embeddings from *word-concept information*. As a first approximation, a concept is a set of semantically similar words. Figure 1 shows an example concept and also indicates one way we learn concepts: *we interpret cliques in the dictionary graph as concepts*. The nodes of the dictionary graph are words, its edges connect words that are translations of each other. A dictionary node has the form prefix:word, e.g., "tpi:wara" (upper left node in the figure). The prefix is the ISO 639-3 code of the language; tpi is Tok Pisin.

Our method takes a parallel corpus as input and induces a dictionary graph from the parallel corpus. Concepts and word-concept pairs are then induced from the dictionary graph. Finally, embeddings are learned from word-concept pairs.

A key application of multilingual embeddings is transfer learning. Transfer learning is mainly of interest if the target is resource-poor. We therefore select as our dataset 1664 translations in 1259 languages of the New Testament from PBC, the Parallel Bible Corpus. Since "translation" is an ambiguous word, we will from now on refer to the 1664 translations as "editions". PBC is aligned

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 1520–1530
Melbourne, Australia, July 15 - 20, 2018.

English King James Version (KJV)	German Elberfelder 1905	Spanish Americas
And he said , Do it the second time . And they did it the second time …	Und er sprach : Füllet vier Eimer mit Wasser , und gießet es auf das Brandopfer und auf das Holz . Und er sprach : Tut es zum zweiten Male ! Und sie taten es zum zweiten Male …	Y dijo : Llenad cuatro cántaros de agua y derramadla sobre el holocausto y sobre la leña . Después dijo : Hacedlo por segunda vez ; y lo hicieron por segunda vez …

Table 1: Instances of verse 11018034. This multi-sentence verse is an example of verse misalignment.

on the verse level; most verses consist of a single sentence, but some contain several (see Table 1). PBC is a good model for resource-poverty; e.g., the training set (see below) of KJV contains fewer than 150,000 tokens in 6458 verses.

We evaluate multilingual embeddings on two tasks, roundtrip translation (RT) and sentiment analysis. RT on the word level is – to our knowledge – a novel evaluation method: a query word w of language L_1 is translated to its closest (with respect to embedding similarity) neighbor v in L_2 and then backtranslated to its closest neighbor w' in L_1. RT is successful if $w = w'$. There are well-known concerns about RT when it is used in the context of machine translation. A successful roundtrip translation does not necessarily imply that v is of high quality and it is not possible to decide whether an error occurred in the forward or backward translations. Despite these concerns about RT on the sentence level, we show that RT on the word level is a difficult task and an effective measure of embedding quality.

Contributions. (i) We introduce a new embedding learning method, multilingual embedding learning through concept induction. (ii) We show that this new concept-based method outperforms previous approaches to multilingual embeddings. (iii) We propose both word-level and character-level dictionary induction methods and present evidence that concepts induced from word-level dictionaries are better for easily tokenizable languages and concepts induced from character-level dictionaries are better for difficult-to-tokenize languages. (iv) We evaluate our methods on a corpus of 1664 editions in 1259 languages. To the best of our knowledge, this is the first detailed evaluation, involving challenging tasks like word translation and crosslingual sentiment analysis, that has been done on such a large number of languages.

2 Methods

2.1 Pivot languages

Most of our methods are based on bilingual dictionary graphs. With 1664 editions, it is computationally expensive to consider all editions simultaneously (more than 10^6 dictionaries). Thus we split the set of editions in 10 pivot and 1654 remaining editions, and do not compute nor use dictionaries within the 1654 editions. We refer to the ten pivot editions as *pivot languages* and give them a distinct role in concept induction. We refer to all editions (including pivot editions) as *target editions*. Thus, a pivot edition has two roles: as a pivot language and as a target edition.

We select the pivot languages based on their sparseness. Sparseness is a challenge in NLP. In the case of embeddings, it is hard to learn a high-quality embedding for any infrequent word. Many of the world's languages (including many PBC languages) exhibit a high degree of sparseness. But some languages suffer comparatively little from sparseness when simple preprocessing like downcasing and splitting on whitespace is employed.

A simple measure of sparseness that affects embedding learning is the number of types. Fewer types is better since their average frequency will be higher. Table 2 shows the ten languages in PBC that have the smallest number of types in 5000 randomly selected verses. We randomly sample 5000 verses per edition and compare the number of types based on this selection because most editions do not contain a few of the selected 6458 verses.

2.2 Character-level modeling (CHAR)

We will see that tokenization-based models have poor performance on a subset of the 1259 languages. To overcome tokenization problems, we represent a verse of length m bytes, as a sequence of $m - (n - 1) + 2$ overlapping byte n-grams. In this paper, "n-gram" always refers to "byte n-gram". We pad the verse with initial and final space, resulting in two additional n-grams (hence "+2"). This representation is in the spirit of earlier byte-level processing, e.g., (Gillick et al., 2016). There are several motivations for this. (i) We can take advantage of byte-level generalizations. (ii) This is robust if there is noise in the byte encoding. (iii) Characters have different properties in different languages and encodings, e.g., English

iso	name	family; (example) region	types	tokens
lhu	Lahu	Sino-Tibetan; Thailand	1452	268
ahk	Akha	Sino-Tibetan; China	1550	315
hak	Hakka Chinese	Chinese; China	1596	242
ium	Iu Mien	Hmong-Mien; Laos	1779	191
tpi	Tok Pisin	Creole; PNG	1815	177
mio	Pinotepa Mixtec	Oto-Manguean; Oaxaca	1828	208
cya	Highland Chatino	Oto-Manguean; Oaxaca	1868	231
bis	Bislama	Creole; Vanuatu	1872	226
aji	Ajië	Austronesian; Houaïlou	1876	194
sag	Sango	Creole; Central Africa	1895	192

Table 2: Our ten pivot languages, the languages in PBC with the lowest number of types. Tokens in 1000s. Tok Pisin and Bislama are English-based and Sango is a Ngbandi-based creole. PNG = Papua New Guinea

Algorithm 1 χ^2-based dictionary induction

procedure DICTIONARYGRAPH(C)
  A = all-edges(C), $E = []$
  for $d \in [1, 2, \ldots, d_{\text{max}}]$ **do**
    $f_{\text{max}} = 2$
    while $f_{\text{max}} \leq |C|$ **do**
      $f_{\text{min}} = \max(\min(5, f_{\text{max}}),\ \frac{1}{10} f_{\text{max}})$
      (χ^2, s, t) = max-χ^2-edge($A, f_{\text{min}}, f_{\text{max}}, d$)
      if $\chi^2 < \chi_{\text{min}}$ **then**
        $f_{\text{max}} = f_{\text{max}} + 1$; continue
      end if
      T = extend-ngram($A, f_{\text{min}}, f_{\text{max}}, d, s, t$)
      append(E, s, T)
      remove-edges(A, s, T)
    end while
  end for
  return dictionary-graph = (nodes(E), E)
end procedure

Figure 2: χ^2-based dictionary induction. C is a sentence-aligned corpus. A is initialized to contain all edges, i.e., the fully connected bipartite graph for each parallel verse. E collects the selected dictionary edges. d is the edge degree: in each pass through the loop only edges are considered whose participating units have a degree less than d. f_{max} is the maximum frequency during this pass. $|C|$ is the number of sentences in the corpus. extend-ngram extends a target ngram to left / right; e.g., if s = "jisas" is aligned with ngram t = "Jesu" in English, then "esus" is added to T. t is always a member of T. remove-edges removes edges in A between s and a member of T.

UTF-8 has properties different from Chinese UTF-8. Thus, universal language processing is easier to design on the byte level.

We refer to this ngram representation as CHAR and to standard tokenization as WORD.

2.3 Dictionary induction

Alignment-based dictionary. We use fastalign (Dyer et al., 2013) to compute word alignments and use GDFA for symmetrization. All alignment edges that occurred at least twice are added to the dictionary graph. Initial experiments indicated that alignment-based dictionaries have poor quality for CHAR, probably due to the fact that overlapping ngram representations of sentences have properties quite different from the tokenized sentences that aligners are optimized for. Thus we use this dictionary induction method only for WORD and developed the following alternative for CHAR.

Correlation-based dictionary (χ^2). χ^2 is a greedy algorithm, shown in Figure 2, that selects, in each iteration, the pair of units that has the highest χ^2 score for cooccurrence in verses. Each selected pair is added to the dictionary and removed from the corpus. Low-frequency units are selected first and high-frequency units last; this prevents errors due to spurious association of high-frequency units with low-frequency units. We perform $d_{\text{max}} = 5$ passes; in each pass, the maximum degree of a dictionary node is $1 \leq d \leq d_{\text{max}}$. So if the node has reached degree d, it is ineligible for additional edges during this pass. Again, this avoids errors due to spurious association of high-frequency units that already participate in many edges with low-frequency units. Recall that this method is only applied for CHAR.

Intra-pivot dictionary. We assume that pivot languages are easily tokenizable. Thus we only consider alignment-based dictionaries (in total 45) within the set of pivot languages.

Pivot-to-target dictionary. We compute an alignment-based and a χ^2-based dictionary between each pivot language and each target edition, yielding a total of 10*1664 dictionaries per dictionary type. (Note that this implies that, for χ^2, the WORD version of the pivot language is aligned with its CHAR version.)

2.4 Concepts

A concept is defined as a set of units that has two subsets: (i) a defining set of words from the ten pivot languages and (ii) a set of target units (words or n-grams) that are linked, via dictionary edges,

Algorithm 2 CLIQUE concept induction

procedure CONCEPTS($I \in \mathbb{R}^{n \times n}, \theta, \nu$)
  $G = ([n], \{(i,j) \in [n] \times [n] \mid I_{ij} > \theta\})$
  cliques = get-cliques($G, 3$)
  $G_c := (V_c, E_c) = (\emptyset, \emptyset)$
  for $c_1, c_2 \in$ cliques $\times$ cliques **do**
    if $|c_1 \cap c_2| \geq \nu \min\{|c_1|, |c_2|\}$ **then**
      $V_c = V_c \cup \{c_1, c_2\}$, $E_c = E_c \cup \{(c_1, c_2)\}$
    end if
  end for
  metacliques = get_cliques($G_c, 1$)
  concepts = $\{$flatten(c) $\mid c \in$ metacliques$\}$
  return concepts
end procedure

Figure 3: CLIQUE concept induction. I is a normalized adjacency matrix of a dictionary graph (i.e., relative frequency of alignment edges with respect to possible alignment edges). get-cliques(G, n) returns all cliques in G of size greater or equal to n. flatten(A) flattens a set of sets. $[n]$ denotes $\{1, 2, \ldots, n\}$. $\theta = 0.4$, $\nu = 0.6$.

to the pivot subset. We selected the ten "easiest" of the 1664 editions as pivot languages. Our premise is that semantic information is encoded in a simply accessible form in the pivot languages and so they should offer a good basis for learning concepts.

We induce concepts from the dictionary graph, a multipartite graph consisting of ten pivot language node/word sets and all target edition node/unit sets (where units are words or n-grams). Edges either connect pivot nodes with other pivot nodes or pivot nodes with target units.

2.4.1 CLIQUE concept induction

If concepts corresponded to each other in the overtly coding pivot languages, if words were not ambiguous and if alignments were perfect, then concepts would be cliques in the pivot part of the dictionary graph. These conditions are too strict for natural languages, so we relax them in our CLIQUE concept induction algorithm (Figure 3). The algorithm identifies maximal multilingual cliques (size ≥ 3) within the dictionary graph of the pivot languages and then merges two cliques if they share enough common words. The merging lets us identify clique-based concepts even if, e.g., a dictionary edge between two words is missing. It also accommodates the situation where more than one word of a pivot language should be part of a concept. The merging step can also be interpreted as metaconcept induction.

Once we have identified the cliques, we project them to the target editions: a target-unit is added to a clique if it is connected to a proportion $\nu = 0.6$ of its member words (to allow for missing edges). This identifies around 150k clique concepts that cover around 8k of the total vocabulary of 24k English words (WORD).

As an alternative to cliques, Ammar et al. (2016) use connected components (CCs). The reachability relation (induced by CC) is the transitive closure of the edge relation. This results in semantically unrelated words being in the same concept for very low levels of noise. In contrast, cliques are more "strict": only node subsets are considered whose corresponding edge relation is already transitive (or almost so for $\nu = 0.6$). Transitivity across languages often does not hold in alignments or dictionaries; see, e.g., Simard (1999). This is why we only consider cliques (which reflect already existent transitivity) rather than CCs, which impose transitivity where it does not hold naturally.

$N(t)$ ={bis:Jorim, ium:yo-lim, sag:Yorim, tpi:Jorim}

$t \in T$={ac0:Yorim,atg0:iJorimu,bav0:Jorim,bom0:Yorim, dik0:Jorim, dtp0:Yorim, duo0:Jorim, engl:Jorim, engb:Jorim, fij2:Lorima, fij3:Jorima, gor0:Yorim, hvn0:Yorim, ibo0:Jorim, iri0:Jorri, kmr0:Yorîm, ksd0:Iorim, kwd0:Jorim, lia0:Yorimi, loz0:Jorimi, mbd0:Hurim, mfh0:Yorim, min0:Yorim, mrw0:Yorim,mse0:Jorimma,naq0:Jorimmi, smo1:Iorimo, srn1:Yorim, tsn2:Jorime, yor2:Jórímù}

Figure 4: Target neighborhood concept example: $N(t) \cup T$. $N(t)$ is the target neighborhood for each of the target words in T.

2.4.2 $N(t)$ (target neighborhood) concept induction

Let $N(t)$ be the neighborhood of target node t in the multipartite dictionary graph, i.e., the set of pivot words that are linked to t. We refer to $N(t)$ as *target neighborhood*. Figure 4 shows an example of such a target neighborhood, the set $N(t)$ consisting of four words.[1] A *target neighborhood concept* consists of a set T of pivot words and all target words t for which $T = N(t)$ holds.

Motivation. Suppose $N(t) = N(u)$ for target nodes t and u from two different languages and $|N(t)|$ covers several pivot languages, e.g., $|N(t)| = |N(u)| = 4$ as in the figure. Again, if units closely corresponded to concepts, if there were no ambiguity, if the dictionary were perfect,

[1] We use numbers and lowercase letters at the fourth position of the prefix to distinguish different editions in the same language, e.g., "0", "3" and "e" in "ace0", "fij3", "enge".

then we could safely conclude that the meanings of t and u are similar; if the meanings of t and u were unrelated, it is unlikely that they would be aligned to the exact same words in four different languages. In reality, there is no exact meaning-form correspondence, there is ambiguity and the dictionary is not perfect. Still, we will see below that defining concepts as target neighborhoods works well.

2.4.3 Filtering target neighborhood concepts

In contrast to CLIQUE, we do not put any constraint on the pivot-to-pivot connections within target neighborhoods; e.g., in Figure 4, we do not require that "bis:Jorim" and "sag:Yorim" are connected by an edge. We evaluate three post-filtering steps of target neighborhoods to increase their quality: restricting target neighborhoods to those that are cliques in $N(t)$**-CLIQUE**; to those that are connected components in $N(t)$**-CC**; and to those of size two that are valid edges in the dictionary in $N(t)$**-EDGE**. For $N(t)$**-EDGE**, we found that taking all edges performs well, so we also consider edges that are proper subsets of target neighborhoods.

2.5 Embedding learning

We adopt the framework of embedding learning algorithms that define contexts and then sample pairs of an input word (more generally, an input unit) and a context word (more generally, a context unit) from each context. The only difference is that our contexts are concepts. For simplicity, we use word2vec (Mikolov et al., 2013a) as the implementation of this model.[2]

2.6 Baselines

Baselines for **multilingual embedding learning.** One baseline is inspired by (Vulić and Moens, 2015). We consider words of one aligned verse in the pivot languages and one target language as a bag of words (BOW) and consider this bag as a context.[3]

Levy et al. (2017) show that sentence ID features (interpretable as an abstract representation of the word's context) are effective. We use a corpus with lines consisting of pairs of an identifier of a verse and a unit extracted from that verse as input to word2vec and call this baseline S-ID.

Lardilleux and Lepage (2009) propose a simple and efficient baseline: **sample-based concept induction**. Words that strictly occur in the same verses are assigned to the same concept. To increase coverage, they propose to sample many different subcorpora.[4] We induce concepts using this method and project them analogous to CLIQUE. We call this baseline SAMPLE.

One novel contribution of this paper is **roundtrip evaluation** of embeddings. We learn embeddings based on a dictionary. The question arises: are the embeddings simply reproducing the information already in the dictionary or are they improving the performance of roundtrip search?

As a baseline, we perform RTSIMPLE, a simple dictionary-based roundtrip translation method. Retrieve the pivot word p in pivot language L_p (i.e., $p \in L_p$) that is closest to the query $q \in L_q$. Retrieve the target unit $t \in L_t$ that is closest to p. Retrieve the pivot word $p' \in L_p$ that is closest to t. Retrieve the unit $q' \in L_q$ that is closest to p'. If $q = q'$, this is an exact hit. We run this experiment for all pivot and target languages.

Note that roundtrip evaluation tests the capability of a system to go from any language to any other language. In an embedding space, this requires two hops. In a highly multilingual dataset of n languages in which not all $O(n^2)$ bilingual dictionaries exist, this requires four hops.

3 Experiments and results

3.1 Data

We use PBC (Mayer and Cysouw, 2014). The version we pulled on 2017-12-11 contains 1664 Bible editions in 1259 languages (based on ISO 639-3 codes) after we discarded editions that have low coverage of the New Testament. We use 7958 verses that have good coverage in these 1664 editions. The data is verse aligned; a verse of the New Testament can consist of multiple sentences. We randomly split verses 6458/1500 into train/test.

3.2 Evaluation

For **sentiment analysis**, we represent a verse as the IDF-weighted sum of its embeddings. Sentiment classifiers (linear SVMs) are trained on the training set of the World English Bible edition

[2] We use code.google.com/archive/p/word2vec

[3] The actual implementation slightly differs to avoid very long lines. It does only consider two pivot languages at a time, but writes each verse multiple times.

[4] We use this implementation: anymalign.limsi.fr

for the two decision problems positive vs. non-positive and negative vs. non-negative. We create a silver standard by labeling verses in English editions with the NLTK (Bird et al., 2009) sentiment classifier.

A positive vs. negative classification is not reasonable for the New Testament because a large number of verses is mixed, e.g., "Now is come salvation ... the power of his Christ: for the accuser ... cast down, which accused them before our God ..." Note that this verse also cannot be said to be neutral. Splitting the sentiment analysis into two subtasks ("contains positive sentiment: yes/no" and "contains negative sentiment: yes/no") is an effective solution for this paper.

The two trained models are then applied to the test set of all 1664 editions. All embeddings in this paper are learned on the training set only. So no test information was used for learning the embeddings.

Roundtrip translation. There are no gold standards for the genre of our corpus (the New Testament); for only a few languages out-of-domain gold standards are available. Roundtrip evaluation is an evaluation method for multilingual embeddings that can be applied if no resources are available for a language. Loosely speaking, for a query q in a query language L_q (in our case English) and a target language L_t, roundtrip translation finds the unit w_t in L_t that is closest to q and then the English unit w_e that is closest to w_t. If the semantics of q and w_e are identical (resp. are unrelated), this is deemed evidence for (resp. counter-evidence against) the quality of the embeddings. We work on the level of Bible edition, i.e., two editions in the same language are considered different "languages".

For a query q, we denote the set of its k_I nearest neighbors in the target edition e by $I_e(q) = \{u_1, u_2, \ldots, u_{k_I}\}$. For each intermediate entry we then consider its k_T nearest neighbors in English. Overall we get a set $T_e(q)$ with $k_I k_T$ predictions for each intermediate Bible edition e. See Figure 5 for an example.

We evaluate the predictions $T_e(q)$ using two sets $G_s(q)$ (strict) and $G_r(q)$ (relaxed) of ground-truth semantic equivalences in English. Precision for a query q is defined as

$$p_i(q) := 1/|E| \sum_{e \in E} \min\{1, |T_e(q) \cap G_i(q)|\}$$

where E is the set of all Bible editions and $i \in \{s, r\}$. We report the mean and median across a set of 70 queries selected from Swadesh (1946)'s list of 100 universal linguistic concepts.

query	intermediate	predictions
woman ⇒	mujer ⇒	wife woman women widows daughters daughter marry married
	esposa ⇒	marry wife woman married marriage virgin daughters bridegroom

Figure 5: Roundtrip translation example for KJV and Americas Bible (Spanish). In this example $\min\{1, |T_e(q) \cap G_i(q)|\}$ equals 0 for S1 and R1, and 1 for S4 and S16.

connu(3), connais(3), connaissent(3), savez(2), sachant(2), sait(2), sachiez(2), savoir, sçai, ignorez, connaissiez, sache connaissez, connaissais, savent, savaient, connoissez, connue, reconnaîtrez, sais, connaissant, savons, connaissait, savait

Figure 6: Intermediates aggregated over 17 French editions. q="know", $N(t)$ embeddings, S16. Intermediates are correct with two possible exceptions: "ignorez" 'you do not know'; "reconnaîtrez" 'you recognize'.

We create G_s and G_r as follows. For WORD, we define $G_s(q) = \{q\}$ and $G_r(q) = L(q)$ where $L(q)$ is the set of words with the same lemma and POS as q. For CHAR, we need to find ngrams that correspond uniquely to the query q. Given a candidate ngram g we consider $c_{qg} := 1/c(g) \sum_{q' \in L(q), \text{substring}(g,q')} c(q')$ where $c(x)$ is the count of character sequence x across all editions in the query language. We add g to $G_i(q)$ if $c_{qg} > \sigma_i$ where $\sigma_s = .75$ and $\sigma_r = .5$. We only consider queries where $G_s(q)$ is non-empty.

We vary the evaluation parameters (i, k_I, k_T) as follows: "S1" represents $(s, 1, 1)$, "S4" $(s, 2, 2)$, "S16" $(s, 2, 8)$, and "R1" $(r, 1, 1)$.

3.3 Corpus generation and hyperparameters

We train with the skipgram model and set vector dimensionality to 200; word2vec default parameters are used otherwise. Each concept – the union of a set of pivot words and a set of target units linked to the pivot words – is written out as a line or (if the set is large) as a sequence of shorter lines. Training corpus size is approximately 50 GB for all experiments. We write several copies of each line (shuffling randomly to ensure lines are different) where the multiplication factor is chosen to result in an overall corpus size of approximately 50 GB.

There are two exceptions. For BOW, we did not find a good way of reducing the corpus size, so this

		roundtrip translation																		sentiment analysis			
		WORD									CHAR									WORD		CHAR	
		S1		R1		S4		S16			S1		R1		S4		S16						
		μ	Md	μ	Md	μ	Md	μ	Md	N	μ	Md	μ	Md	μ	Md	μ	Md	N	pos	neg	pos	neg
1	RTSIMPLE	33	24	37	36					67	24	13	32	21					70				
2	BOW	7	5	8	7	13	12	26	28	69	3	2	3	2	5	4	10	11	70	33	81	13	83
3	S-ID	46	46	52	55	63	76	79	91	65	9	5	9	5	14	9	25	22	70	79	88	65	86
4	SAMPLE	33	23	43	42	54	59	82	96	65	53	**59**	59	**72**	67	85	79	99	58	82	89	77	89
5	CLIQUE	43	36	59	63	67	77	93	99	69	42	46	48	55	60	76	73	98	53	84	89	69	88
6	N(t)	**54**	**59**	**61**	69	**80**	**87**	**94**	**100**	69	50	53	54	59	73	82	90	99	66	82	89	**87**	**90**
7	N(t)-CLIQUE	11	0	11	0	16	0	22	0	18	39	45	41	47	58	74	76	94	56	22	84	61	84
8	N(t)-CC	3	0	3	0	5	0	7	0	5	11	0	11	0	16	0	25	0	21	4	84	40	83
9	N(t)-EDGE	35	30	43	36	56	55	87	94	69	39	29	49	52	64	78	88	**100**	63	84	**90**	84	89

Table 3: Roundtrip translation (mean/median accuracy) and sentiment analysis (F_1) results for word-based (WORD) and character-based (CHAR) multilingual embeddings. N (coverage): # queries contained in the embedding space. The best result *across WORD and CHAR* is set in bold.

corpus is 10 times larger than the others. For S-ID, we use Levy et al. (2017)'s hyperparameters; in particular, we trained for 100 iterations and we wrote each verse-unit pair to the corpus only once, resulting in a corpus of about 4 GB.

We set the n parameter of n-grams to $n = 4$ for Bible editions with $\rho < 2$, $n = 8$ for Bible editions with $2 \leq \rho < 3$ and $n = 12$ for Bible editions with $\rho \geq 3$ where ρ is the ratio between size in bytes of the edition and median size of the 1664 editions. In χ^2 dictionary induction, we set $\chi_{\text{min}} = 100$. In the concept induction algorithm we set $\theta = 0.4$ and $\nu = 0.6$. Except for SAMPLE and CLIQUE, we filter out hapax legomena.

3.4 Results

Table 3 presents evaluation results for roundtrip translation and sentiment analysis.

Validity of roundtrip (RT) evaluation results. RTSIMPLE (line 1) is not competitive; e.g., its accuracy is lower by almost half compared to $N(t)$. We also see that RT is an excellent differentiator of poor multilingual embeddings (e.g., BOW) vs. higher-quality ones like S-ID and $N(t)$. This indicates that RT translation can serve as an effective evaluation measure.

The **concept-based multilingual embedding learning** algorithms CLIQUE and $N(t)$ (lines 5-6) consistently (except S1 WORD) outperform BOW and S-ID (lines 2-3) that are not based on concepts. BOW performs poorly in our low-resource setting; this is not surprising since BOW methods rely on large datasets and are therefore expected to fail in the face of severe sparseness. S-ID performs reasonably well for WORD, but even in that case it is outperformed by $N(t)$, in some cases by a large margin, e.g., μ of 63 for S-ID vs. 80 for $N(t)$ for S4. For CHAR, S-ID results are poor. On sentiment classification, $N(t)$ also consistently outperforms S-ID.

While S-ID provides a clearer signal to the embedding learner than BOW, it is still relatively crude to represent a word as – essentially – its binary vector of verse occurrence. Concept-based methods perform better because they can exploit the more informative dictionary graph.

Comparison of graph-theoretic definitions of concepts: $N(t)$-CLIQUE, $N(t)$-CC. $N(t)$ (line 6) has the most consistent good performance across tasks and evaluation measures. Postfiltering target neighborhoods down to cliques (line 7) and CCs (line 8) does not work. The reason is that the resulting number of concepts is too small; see, e.g., low coverages of $N = 18$ ($N(t)$-CLIQUE) and $N = 5$ ($N(t)$-CC) for WORD and $N = 21$ ($N(t)$-CC) for CHAR. $N(t)$-CLIQUE results are highly increased for CHAR, but still poorer by a large margin than the best methods. We can interpret this result as an instance of a precision-recall tradeoff: presumably the quality of the concepts found by $N(t)$-CLIQUE and $N(t)$-CC is better (higher precision), but there are too few of them (low recall) to get good evaluation numbers.

Comparison of graph-theoretic definitions of concepts: CLIQUE. CLIQUE has strong performance for a subset of measures, e.g., ranks consistently second for RT (except S1 WORD) and sentiment analysis in WORD. Although CLIQUE is perhaps the most intuitive way of inducing a concept from a dictionary graph, it may suffer in relatively high-noise settings like ours.

Comparison of graph-theoretic definitions of concepts: $N(t)$ vs. $N(t)$-EDGE. Recall that $N(t)$-EDGE postfilters target neighborhoods by

[ksw] ဒီးတာ်ကမၤဉ်လၢအဟ်လၢယလိၤခဲကနံၤဉ်အံၤ⋆,
⋆ထုးပှ်ၤအိၤဉ်အသးတနှ်ဘၤဉ်⋆.
[cso] Hi³⋆sa³jun³⋆lɨ́¹³⋆ma³tson²⋆tsú²⋆
lɨ³ua³⋆cáun²⋆tso³⋆ñí¹⋆hná¹⋆ní²⋆.
[eng] Neither⋆can⋆they⋆prove⋆the⋆things⋆
whereof⋆they⋆now⋆accuse⋆me⋆.

Figure 7: Verse 44024013. "*" = tokenization boundary. S'gaw Karen (ksw) is difficult to tokenize and CHAR > WORD for $N(t)$. Chinanteco de Sochiapan (cso) has few types, similar to a pivot language, and CHAR < WORD for $N(t)$.

N(t) [CHAR]		S-ID [WORD]		SAMPLE [WORD]		CLIQUE [WORD]	
iso	Δ	iso	Δ	iso	Δ	iso	Δ
arb1	54	pua0	61	jpn1	42	mya2	38
arz0	53	sun2	54	khm2	40	jpn1	36
cop3	49	jpn1	53	cap2	40	khm3	34
srp0	44	khm3	53	khm3	40	bsn0	28
cop2	44	khm2	50	mya2	39	khm2	27
...	...	...	...	...	...	...	...
pis0	-23	vie7	-24	eng8	-7	haw0	-22
pcm0	-23	kri0	-25	enm1	-9	eng4	-23
ksw0	-24	tdt0	-27	lzh2	-9	enm2	-26
lzh2	-41	eng2	-27	eng4	-12	enm1	-26
lzh1	-51	vie6	-29	lzh1	-13	engj	-28

Table 4: Comparison of $N(t)$[WORD] with four other methods. Difference in mean performance (across queries) in R1 per edition. Positive number means better performance of $N(t)$[WORD].

only considering pairs of pivot words that are linked by a dictionary edge. This "quality" filter does seem to work in some cases, e.g., best performance S16 Md for CHAR. But results for WORD are much poorer.

SAMPLE performs best for CHAR: best results in five out of eight cases. However, its coverage is low: $N = 58$. This is also the reason that it does not perform well on sentiment analysis for CHAR ($F_1 = 77$ for pos).

Target neighborhoods $N(t)$. The overall best method is $N(t)$. It is the best method more often than any other method and in the other cases, it ranks second. This result suggests that the assumption that two target units are semantically similar if they have dictionary edges with exactly the same set of pivot words is a reasonable approximation of reality. Postfiltering by putting constraints on eligible sets of pivot words (i.e., the pivot words themselves must have a certain dictionary link structure) does not consistently improve upon target neighborhoods.

WORD vs. CHAR. For roundtrip, WORD is a better representation than CHAR if we just count the bold winners: seven (WORD) vs. three (CHAR), with two ties. For sentiment, the more difficult task is pos and for this task, CHAR is better by 3 points than WORD ($F_1 = 87$, line 6, vs. $F_1 = 84$, lines 9/5). However, Table 4 shows that CHAR < WORD for one subset of editions (exemplified by cso in Figure 7) and CHAR > WORD for a different subset (exemplified by ksw). So there are big differences between CHAR and WORD in both directions, depending on the language. For some languages, WORD performs a lot better, for others, CHAR performs a lot better.

We designed RT evaluation as a word-based evaluation that disfavors CHAR in some cases. The fourgram "ady@" in the World English Bible occurs in "already" (32 times), "ready" (31 times) and "lady" (9 times). Our RT evaluation thus disqualifies "ady@" as a strict match for "ready". But all 17 *aligned* occurrences of "ady@" are part of "ready" – all others were not aligned. So in the χ^2-alignment interpretation, $P(\text{ready}|\text{ady@}) = 1.0$. In contrast to RT, we only used aligned ngrams in the sentiment evaluation. This discrepancy may explain why the best method for sentiment is a CHAR method whereas the best method for RT is a WORD method.

First NLP task evaluation on more than 1000 languages. Table 3 presents results for 1664 editions in 1259 languages. To the best of our knowledge, this is the first detailed evaluation, involving two challenging NLP tasks, that has been done on such a large number of languages. For several methods, the results are above baseline for all 1664 editions; e.g., S1 measures are above 20% for all 1664 editions for $N(t)$ on CHAR.

4 Related Work

Following Upadhyay et al. (2016), we group **multilingual embedding** methods into classes A, B, C, D.

Group A trains monolingual embedding spaces and subsequently uses a transformation to create a unified space. Mikolov et al. (2013b) find the transformation by minimizing the Euclidean distance between word pairs. Similarly, Zou et al. (2013), Xiao and Guo (2014) and Faruqui and Dyer (2014) use different data sources for identifying word pairs and creating the transformation (e.g., by CCA). Duong et al. (2017) is also simi-

lar. These approaches need large datasets to obtain high quality monolingual embedding spaces and are thus inappropriate for a low-resource setting of 150,000 tokens per language.

Group B starts from the premise that representation of aligned sentences should be similar. Neural network approaches include (Hermann and Blunsom, 2014a) (BiCVM) and (Sarath Chandar et al., 2014) (autoencoders). Again, we have not enough data for training neural networks of this size. Søgaard et al. (2015) learn an interlingual space by using Wikipedia articles as concepts and applying inverted indexing. Levy et al. (2017) show that what we call S-ID is a strongly performing embedding learning method. We use S-ID as a baseline.

Group C combines mono- and multilingual information in the embedding learning objective. Klementiev et al. (2012) add a word-alignment based term in the objective. Luong et al. (2015) extend Mikolov et al. (2013a)'s skipgram model to a bilingual model. Gouws et al. (2015) introduce a crosslingual term in the objective, which does not rely on any word-pair or alignment information. For n editions, including $O(n^2)$ bilingual terms in the objective function does not scale.

Group D creates pseudocorpora by merging data from multiple languages into a single corpus. One such method, due to Vulić and Moens (2015), is our baseline BOW.

Östling (2014) generates **multilingual concepts** using a Chinese Restaurant process, a computationally expensive method. Wang et al. (2016) base their concepts on cliques. We extend their notion of clique from the bilingual to the multilingual case. Ammar et al. (2016) use connected components. Our baseline SAMPLE, based on (Lardilleux and Lepage, 2007, 2009), samples aligned sentences from a multilingual corpus and extracts perfect alignments.

Malaviya et al. (2017), Asgari and Schütze (2017), Östling and Tiedemann (2017) and Tiedemann (2018) perform **evaluation** on the language level (e.g., typology prediction) for 1000+ languages or perform experiments on 1000+ languages without evaluating each language. We present the first work that evaluates on 1000+ languages on the sentence level on a difficult task.

Somers (2005) criticizes RT evaluation on the sentence level; but see Aiken and Park (2010). We demonstrated that when used on the word/unit level, it distinguishes weak from strong embeddings and correlates well with an independent sentiment evaluation.

Any alignment algorithm can be used for **dictionary induction**. We only used a member of the IBM class of models (Dyer et al., 2013), but presumably we could improve results by using either higher performing albeit slower aligners or non-IBM aligners (e.g., (Och and Ney, 2003; Tiedemann, 2003; Melamed, 1997)). Other alignment algorithms include 2D linking (Kobdani et al., 2009), sampling based methods (e.g., Vulic and Moens (2012)) and EFMARAL (Östling and Tiedemann, 2016). EFMARAL is especially intriguing as it is based on IBM1 and Agić et al. (2016) find IBM2-based models to favor closely related languages more than models based on IBM1. However, the challenge is that we need to compute tens of thousands of alignments, so speed is of the essence. We ran character-based and word-based induction separately; combining them is promising future research; cf. (Heyman et al., 2017).

There is much work on embedding learning that does not require **parallel corpora**, e.g., (Vulić and Moens, 2012; Ammar et al., 2016). This work is more generally applicable, but a parallel corpus provides a clearer signal and is more promising (if available) for low-resource research.

5 Summary

We presented a new method for estimating vector space representations of words: embedding learning by concept induction. We tested this method on a highly parallel corpus and learned semantic representations of words in 1259 different languages in a single common space. Our extensive experimental evaluation on crosslingual word similarity and sentiment analysis indicates that concept-based multilingual embedding learning performs better than previous approaches.

The embedding spaces of the 1259 languages (SAMPLE, CLIQUE and $N(t)$) are available: http://cistern.cis.lmu.de/comult/.

We gratefully **acknowledge** funding from the European Research Council (grants 740516 & 640550) and through a Zentrum Digitalisierung.Bayern fellowship awarded to the first author. We are indebted to Michael Cysouw for making PBC available to us.

References

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

Milam Aiken and Mina Park. 2010. The efficacy of round-trip translation for MT evaluation. *Translation Journal*, 14(1).

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

Ehsaneddin Asgari and Hinrich Schütze. 2017. Past, present, future: A computational investigation of the typology of tense in 1000 languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly Media.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2017. Multilingual training of crosslingual word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Karl Moritz Hermann and Phil Blunsom. 2014a. Multilingual distributed representations without word alignment. In *Proceedings of the 2014 International Conference on Learning Representations*.

Karl Moritz Hermann and Phil Blunsom. 2014b. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

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*.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. *Proceedings of the 24th International Conference on Computational Linguistics*.

Hamidreza Kobdani, Alex Fraser, and Hinrich Schütze. 2009. Word alignment by thresholded two-dimensional normalization. In *Proceeedings of the 12th Machine Translation Summit*.

Adrien Lardilleux and Yves Lepage. 2007. The contribution of the notion of hapax legomena to word alignment. In *Proceedings of the 4th Language and Technology Conference*.

Adrien Lardilleux and Yves Lepage. 2009. Sampling-based multilingual alignment. In *Proceedings of 7th Conference on Recent Advances in Natural Language Processing*.

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*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel bible corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*.

Ryan T. McDonald, Slav Petrov, and Keith B. Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

I. Dan Melamed. 1997. A word-to-word model of translational equivalence. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*.

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

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

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Robert Östling. 2014. Bayesian word alignment for massively parallel texts. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

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

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

AP Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh M. Khapra, Balaraman Ravindran, Vikas C. Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In *Proceedings of the 2014 Annual Conference on Neural Information Processing Systems*.

Michel Simard. 1999. Text-translation alignment: Three languages are better than two. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual nlp. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing*.

Harold Somers. 2005. Round-trip translation: What is it good for? In *Proceedings of the Australasian Language Technology Workshop 2005*.

Morris Swadesh. 1946. South Greenlandic (Eskimo). In Cornelius Osgood, editor, *Linguistic Structures of Native America*. Viking Fund Inc. (Johnson Reprint Corp.), New York.

Jörg Tiedemann. 2003. Combining clues for word alignment. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*.

Jörg Tiedemann. 2018. Emerging language spaces learned from massively multilingual corpora. *arXiv preprint arXiv:1802.00273*.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Ivan Vulić and Marie-Francine Moens. 2012. Detecting highly confident word translations from comparable corpora without any prior knowledge. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*.

Ivan Vulic and Marie-Francine Moens. 2012. Subcorpora sampling with an application to bilingual lexicon extraction. In *Proceedings of the 24th International Conference on Computational Linguistics*.

Ivan Vulić 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 and the 7th International Joint Conference on Natural Language Processing*, volume 2.

Rui Wang, Hai Zhao, Sabine Ploux, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2016. A novel bilingual word embedding method for lexical translation using bilingual sense clique. *arXiv preprint arXiv:1607.08692*.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the 18th Conference on Computational Natural Language Learning*.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*.

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*.

Chapter 3

Multilingual Embeddings Jointly Induced from Contexts and Concepts: Simple, Strong and Scalable

Multilingual Embeddings Jointly Induced from Contexts and Concepts: Simple, Strong and Scalable

Philipp Dufter, Mengjie Zhao, Hinrich Schütze
Center for Information and Language Processing (CIS), LMU Munich, Germany
philipp@cis.lmu.de, mengjie.zhao@cis.lmu.de

Abstract

Word embeddings induced from *local context* are prevalent in NLP. A simple and effective context-based multilingual embedding learner is Levy et al. (2017)'s S-ID (sentence ID) method. Another line of work induces high-performing multilingual embeddings from *concepts* (Dufter et al., 2018). In this paper, we propose Co+Co, a simple and scalable method that combines context-based and concept-based learning. From a sentence-aligned corpus, concepts are extracted via sampling; words are then associated with their concept ID and sentence ID in embedding learning. This is the first work that successfully combines context-based and concept-based embedding learning. We show that Co+Co performs well for two different application scenarios: the Parallel Bible Corpus (1000+ languages, low-resource) and EuroParl (12 languages, high-resource). Among methods applicable to both corpora, Co+Co performs best in our evaluation setup of six tasks.

1 Introduction

Multilingual embeddings are useful because they provide word representations of source and target language in the same space in machine translation and because they are a basis for transfer learning. In contrast to prior multilingual work (Zeman and Resnik, 2008; McDonald et al., 2011; Tsvetkov et al., 2014), automatically learned embeddings potentially perform as well but are more efficient and easier to use (Klementiev et al., 2012; Hermann and Blunsom, 2014b; Guo et al., 2016). Thus, multilingual word embedding learning is important for natural language processing (*NLP*).

The quality of multilingual embeddings is driven by the underlying feature set more than the type of algorithm used for training the embeddings (Upadhyay et al., 2016; Ruder et al., 2019). Most embedding learners build on using *context information* as feature. Dufter et al. (2018) showed that using *concept information* is effective for multilingual embedding learning, as well. We propose *Co+Co*, a method that combines the concept identification method *Anymalign* (Lardilleux and Lepage, 2009) and the multilingual embedding learning method sentence ID (*S-ID*) (Levy et al., 2017) into a embedding learning method that is based on both concept and context.

Our aim is to create a method for learning non-contextualized embeddings that yield strong results while being scalable and widely applicable. Thus we work on two parallel corpora: a low-resource, massively multilingual corpus, the Parallel Bible Corpus (*PBC*), and on a high-resource, mildly multilingual corpus, *EuroParl*. Out of 15 embedding learning methods we identify S-ID, the concept based method N(t) by Dufter et al. (2018) and Co+Co as the only ones that yield high-quality word spaces across corpora. Co+Co exhibits the best and most stable performance.

Our contributions are: **i)** We show that Co+Co, i.e., using concepts and contexts jointly, yields higher quality embeddings than either by itself. **ii)** We demonstrate that Co+Co performs well across very different datasets and scales across a high number of languages. **iii)** We find that lower embedding dimensionality is better for word translation in PBC. In addition, we find that QVEC evaluation (Tsvetkov et al., 2015) is highly dependent on dimensionality.

2 Methods

Our proposed method consists of three steps: 1) Inducing concepts using Anymalign (Lardilleux and Lepage, 2009). 2) Generating an artificial corpus using sentence and concept IDs. 3) Training word2vec on the artificial corpus.

2.1 Concept Induction

Lardilleux and Lepage (2009) propose Anymalign, an algorithm originally intended for obtaining word alignments. Consider a parallel corpus V across multiple languages. The central idea is that words that occur strictly in the same sentences, can be considered translations. In addition to words, word ngrams can be considered. We call words or word ngrams that occur exclusively in the same sentences *perfectly aligned*. By this strict definition, the number of perfect alignments is low. Coverage can be increased by sampling subcorpora. As the number of sentences is smaller in each sample and there is a high number of sampled subcorpora, perfect alignments occur more often.

Figure 1 shows Lardilleux and Lepage (2009)'s Anymalign algorithm on a high level.

There are three relevant hyperparameters in Anymalign: the minimum number of languages a perfect alignment should cover (*MinLg.*) and the maximum ngram length (*MaxNgr.*). The size of a subsample is adjusted automatically to maximize the probability that each sentence is sampled at least once. This probability depends on the number of samples drawn and thus on the runtime (T) of the algorithm. Thus, T is another hyperparameter. For details see (Lardilleux and Lepage, 2009).

We argue that with small MaxNgr. one can interpret perfect alignments as concepts, i.e., a set of semantically similar words. For example the English trigram "mount of olives" and the French trigram "montagne des oliviers" are a perfect alignment describing the same concept. Thus we define a *concept* as a set of perfectly aligned words and use Anymalign as concept induction algorithm. Note that most members of a concept are not perfect alignments in V (only in a subsample V') and that a word can be part of multiple concepts. See Section 5 for comments on concept quality.

2.2 Corpus Creation

We use sentence IDs from the parallel corpus and concepts to create corpora. Figure 2 shows samples of the generated corpora to be processed by the embedding learner.

S-ID. We adopt Levy and Goldberg (2014)'s framework; it formalizes the basic information that is passed to the embedding learner as a set of pairs. In the monolingual case, each pair consists of two words that occur in the same context. A successful approach to multilingual embedding learning for parallel corpora is to use pairs of a word and a sentence ID (Levy et al., 2017). The sentence ID acts as crosslingual signal. We call this method *S-ID*.[1]

C-ID. We propose to adjust S-ID by replacing sentence IDs with concept IDs. That is, a line consists of a concept ID and a concept member word.

Co+Co. We combine S-IDs and C-IDs by creating two corpora with the respective method and then concatenating their corpora before learning embeddings. Intuitively, both methods use complementary information: S-ID uses local context information and C-ID leverages globally aggregated information. Therefore, we expect a higher performance by combining those.

Algorithm 1 Anymalign (Lardilleux and Lepage, 2009)

```
procedure GetConcepts(V, MinLg., MaxNgr., T)
    C = ∅
    while runtime ≤ T do
        V' = get-subsample(V)
        A = get-concepts(V')
        A = filter-concepts(A, MinLg., MaxNgr.)
        C = C ∪ A
    end while
end procedure
```

Figure 1: V is a parallel corpus. *get-subsample* creates a sentence-aligned parallel subcorpus by sampling lines from V. *get-concepts* returns the set of perfect alignments (concepts). *filter-concepts* filters the set of concepts to enforce MinLg. and MaxNgr.

```
[...]                                [...]
48001018 enge:fifteen                C:911 kqc0:Jerusalem
48001018 enge:,                      C:911 por5:Jerusalém
48001018 enge:years                  C:911 eng7:Jerusalem
48001018 deu0:fünfzehn               C:911 haw0:Ierusalema
48001018 deu0:Jahre                  C:911 ilb0:Jelusalemu
48001018 deu0:,                      C:911 fra1:Jérusalem
[...]                                [...]
```

```
[...]
45016016 Salute one another with an holy kiss . The churches of
         Christ salute you .
48001018 Then after three years I went up to Jerusalem to see Peter
         , and abode with him fifteen days .
[...]
```

Figure 2: Samples of S-ID (top-left) and C-ID (top-right) corpora that are input to word2vec. Words are prefixed by a 3 character ISO 639-3 language identifier followed by an alphanumeric character to distinguish editions in the same language (e.g., enge = English King James Version (James-Ed.)). C:911 is a concept identifier. Bottom: James-Ed. text sample.

[1] Note that we use the name S-ID for the sentence identifier, for the corpus creation method that is based on these identifiers, for the embedding learning method based on such corpora and for the embeddings produced by the method. The same applies to other method names. Which sense is meant should be clear from context.

2.3 Embedding Learning

To obtain embeddings we use use *word2vec skip-gram*[2] (Mikolov et al., 2013a) on the generated corpora. We use default hyperparameters, and investigate three more closely: number of iterations (*NIter.*), minimum frequency of a word (*MinCount.*) and embedding dimensionality (*Dim.*). Throughout the paper we ℓ^2-normalize vectors.

3 Application to Parallel Bible Corpus

3.1 Data

We work on *PBC*, the Parallel Bible Corpus (Mayer and Cysouw, 2014), a verse-aligned corpus of 1000+ translations of the New Testament. For the sake of comparability we use the same 1664 Bible editions across 1259 languages (distinct ISO 639-3 codes) and the same 6458 training verses as in (Dufter et al., 2018).[3] We follow their terminology and refer to "translations" as "editions". PBC is a good model for resource-poverty; e.g., the training set of the English King James Version (*James-Ed.*) contains fewer than 150,000 tokens in 6458 verses. James-Ed. spans a vocabulary of 6162 words and all 32 English editions together cover 23,772 words. We use the tokenization provided in the data, which is erroneous for some hard-to-tokenize languages (e.g., Khmer, Japanese), and do not apply further preprocessing.

3.2 Evaluation

Dufter et al. (2018) introduce roundtrip translation (*RTT*) as multilingual embedding evaluation when no gold standard is available. A query word q in language L_1 is translated to its nearest neighbor (by cosine similarity) v in an intermediate language L_2 and then backtranslated to its closest neighbor q' in the query language L_1. RTT is successful if $q = q'$. For the roundtrips we consider k_I nearest neighbors in L_2 and, for each of these intermediate neighbor, k_T predictions in L_1.

Predictions are compared to a ground truth set G_q. There is a strict ($\{q\}$) and a relaxed ($\{$words with the same lemma and part-of-speech as $q\}$) ground truth. Thus multiple roundtrips of q can be considered correct (e.g., inflections of a query). We average binary results (per query) over editions and report mean (μ) and median (*Md.*) over queries. Inspired by the precision@k evaluation for word translation we vary (k_I, k_T) as follows: "S1" $(1,1)$, "R1" $(1,1)$, "S4" $(2,2)$, and "S16" $(2,8)$, where S (R) stand for using the strict (relaxed) ground truth.

If q is not in the embedding space, we consider the roundtrip as failed. The number of queries contained in the embedding space is denoted by N ("coverage"). We use the same queries as in (Dufter et al., 2018), which are based on (Swadesh, 1946)'s 100 universal words. In addition, we introduce a development set: an earlier list by Swadesh with 215 words.[4] 151 queries remain in the development set after excluding queries from the test set. Due to this large number we do not compute the relaxed measure on the development set as this requires manual effort on the ground truth. We work on the level of Bible editions, i.e., two editions in the same language are considered different "languages". We use James-Ed. as the query edition if James-Ed. contains q. Else we randomly choose another English edition.

As extrinsic task we perform sentiment analysis following (Dufter et al., 2018). We use their ground truth and data split into training and test set. There are two classifications: whether a verse contains positive (*Pos.*) or negative (*Neg.*) sentiment. Given the multilingual space, for each task a linear support vector machine (*SVM*) is trained on the English World Edition (*World-Ed.*) (following (Dufter et al., 2018)) and subsequently tested on all other editions. We train SVMs in 5-fold cross-validation on the training set to optimize the hyperparameter C. We report average F_1 across editions.

3.3 Baselines

We compute diverse baselines to pinpoint reasons for performance changes as much as possible.

Monolingual Embedding Space. In the baseline *MONO* we train 1664 monolingual spaces using word2vec and interpret them as if they were a shared multilingual embedding space. This serves as consistency check for our evaluation methods.

Context Based Embedding Space. We use *S-ID* to obtain a multilingual space.

Transformation Based Embedding Space. The baseline *LINEAR* follows (Duong et al., 2017). We pick one edition as the "embedding space defining edition", in our case the English Catholic Bible (*Cath-Ed.*). We did not choose

[2] We use code.google.com/archive/p/word2vec

[3] Information downloaded from cistern.cis.lmu.de/comult

[4] concepticon.clld.org/contributions/Swadesh-1950-215

James-Ed. or World-Ed. to avoid that one single edition has multiple special roles. We then create 1664 bilingual embedding spaces using S-ID; in each case the two editions covered are Cath-Ed. and one of the other 1664 languages. We then use (Mikolov et al., 2013b)'s linear transformation method to map all embedding spaces to the Cath-Ed. embedding space. More specifically, let $X' \in \mathbb{R}^{n_X \times d}$, $Y' \in \mathbb{R}^{n_Y \times d}$ be two embedding spaces, with n_X, n_Y number of words, and d be the embedding dimension. We then select n_T transformation words that are contained in both embedding spaces. This yields: $X, Y \in \mathbb{R}^{n_T \times d}$. The transformation matrix $W \in \mathbb{R}^{d \times d}$ is then given by $\arg\min_W \|XW - Y\|_F$ where $\|\cdot\|_F$ is the Frobenius norm. The closed form solution is given by $W^* = X^+ Y$ where X^+ is the Moore-Penrose Pseudoinverse (Penrose, 1956). In our case, X is a bilingual and Y is a monolingual embedding space, both containing the vocabulary of Cath-Ed.

Bilingual Embedding Spaces. *BILING* uses the same bilingual embedding spaces as LINEAR, but we do not transform the bilingual spaces into a common space. This baseline shows the effect of multilingual vs. bilingual embedding spaces. As we do not have a multilingual space we need to modify our evaluation methods slightly: we perform RTT in each bilingual embedding space separately. For sentiment analysis, we train one SVM per embedding space on English, which is then tested on the other edition.

Unsupervised Embedding Learning. We apply the recent unsupervised embedding learning method by Lample et al. (2018) (*MUSE*).[5] Given unaligned corpora in two languages, MUSE learns two separate embedding spaces that are subsequently unified by a linear transformation. This transformation is learned using a discriminator neural network that tries to identify the original language of a vector. Again, we learn monolingual embedding spaces by running word2vec on PBC directly. Subsequently, we transform all spaces into the word space of Cath-Ed. using MUSE. Chen and Cardie (2018) extended MUSE multilingually. We include their method *MAT+MPSR* as baseline. MAT+MPSR is memory and computation intensive. Allocating three days of computation on a standard GPU (GTX 1080 Ti), we were only able to apply this baseline on a subset of 52 editions.

[5]We use https://github.com/facebookresearch/MUSE

		NIter.	MinCount.	Dim.	S1 μ	S1 Md	S4 μ	S4 Md	S16 μ	S16 Md	N
1	S-ID	**100**	5	200	29	21	43	46	56	78	103
2	S-ID	5			14	11	25	22	41	45	103
3	S-ID	10			25	16	38	34	51	60	103
4	S-ID	25			27	20	41	43	53	69	103
5	S-ID	50			27	16	40	40	53	67	103
6	S-ID	150			29	21	43	47	56	79	103
7	S-ID		**2**		**35**	**31**	**52**	**60**	**66**	**90**	130
8	S-ID		10		24	5	36	17	48	63	85
9	S-ID			**100**	30	24	45	48	58	83	103
10	S-ID			300	28	19	42	41	54	69	103

		MinLg.	MaxNgr.	T	S1 μ	S1 Md	S4 μ	S4 Md	S16 μ	S16 Md	N
1	C-ID	**100**	**3**	10	**30**	**26**	**46**	**46**	60	71	120
2	C-ID	50			26	21	39	42	56	**79**	104
3	C-ID	150			22	14	34	28	47	56	94
4	C-ID	500			15	0	25	0	33	0	59
5	C-ID		1		17	0	27	0	38	0	73
6	C-ID		5		28	20	40	35	55	58	122
7	C-ID			5	24	20	36	34	48	52	114
8	C-ID			**15**	**30**	**26**	**46**	44	**61**	76	121

Table 1: Hyperparameter selection for word2vec (top) and Anymalign (bottom) on RTT. Initial parameters in first row; empty cell: initial parameter from the first row. For example: to compare the effect of different dimensionality in word2vec compare lines 1, 9 and 10 in the top table (best dimension is 100). Bold: best result per column or selected hyperparameter value.

Non-Embedding Baseline. To show that embedding spaces provide some advantages over using the concepts as is, we introduce *C-SIMPLE*, a non-embedding baseline that follows the idea of RTT. Given a query word q and an intermediate edition, we consider all words that share a concept ID with q as possible intermediate words. We then choose randomly (probability weights according to number of concepts shared with q) intermediate words. For back translation we apply the same procedure.

3.4 Hyperparameter Selection

We select hyperparameters based on the roundtrip translation task.

Word2vec. We tune word2vec parameters based on the method S-ID. Since a grid search for optimal values for the parameters NIter., MinCount. and Dim. would take too long, we search greedily instead: we choose an initial parameter setting, vary one parameter at a time and select the value with the best performance. More iterations yield better performance. NIter. = 100 is a good efficiency-performance trade-off. For MinCount. (minimum frequency of a word) the best performance is found using 2. This is mainly due to

		Roundtrip									Sentiment	
		S1		R1		S4		S16				
		μ	Md	μ	Md	μ	Md	μ	Md	N	Pos.	Neg.
1	C-SIMPLE	35	33	35	34	49	54	56	56	67		
2	MONO	21	18	21	18	39	36	67	68	69	4	74
3	S-ID	48	47	53	59	65	72	83	93	69	71	88
4	C-ID	43	42	46	43	58	60	79	91	67	73	86
5	Co+Co	51	50	56	65	69	80	86	96	69	79	**89**
6	LINEAR	30	28	31	30	43	46	62	71	69	76	83
7	MUSE	16	13	16	13	33	28	65	64	69	13	73
8	MAT+MPSR	11	10	11	10	20	17	44	42	66	37	72
9	BILING	41	32	42	35	53	55	67	82	69	80	87
10	N(t)*	**54**	**59**	**61**	**69**	**80**	**87**	**94**	**100**	69	**82**	**89**
11	SAMPLE*	33	23	43	42	54	59	82	96	65	**82**	**89**

Table 2: Evaluation results. *: results by (Dufter et al., 2018). Sentiment analysis not defined for C-Simple. Bold: best result per column, Underlined: second best.

increased coverage. Surprisingly, smaller embedding dimensions work better to some degree. A highly multilingual embedding space is expected to suffer more from ambiguity and that is an argument for higher dimensionality; cf. Li and Jurafsky (2015). But this effect seems to be counteracted by the low-resource properties of PBC for which the increased number of parameters of higher dimensionalities cannot be estimated reliably. We choose embedding size 100.

Anymalign. We tune the hyperparameters for Anymalign by evaluation embedding spaces created with C-ID. We use the above word2vec settings except for MinCount. As argued before, each word in a concept carries a strong multilingual signal, which is why we do not apply any frequency filtering for C-ID. Thus we set MinCount. to 1 whenever we learn concept based embeddings. For Co+Co we apply the different frequency thresholds on the two parts (S-ID and C-ID parts) of the corpus.

We find the best performance when setting the minimum number of editions (MinLg.) to 100. As expected coverage worsens when increasing MinLg. MaxNgr. is the maximum ngram length. We see the best performance when setting MaxNgr. to 3. This is intuitive for languages like Swedish since compounds can be found. T is the time in hours, i.e., for how long to sample (which steers the size and number of sampled subcorpora). As expcted higher T yields better performance. Given only a slight difference between 10 and 15 we set T to 15 for slightly higher coverage.

3.5 Results

Table 2 presents results on the test set.

Embeddings vs. Concepts. C-Simple works reasonably well but is outperformed by most embedding spaces. This indicates that learning embeddings augments the concept information.

Evaluation Difficulty. MONO is the weakest baseline and is a good indicator that the RTT task is challenging, e.g., S1=21 and R1=18. It admits the weakness of RTT by yielding a non-zero performance despite most of the intermediate neighbours being not related to the query at all. Still it proves that RTT is a good indicator for the quality of multilingual word spaces, as truly multilingual word spaces significantly outperform MONO (especially for the median). The same is true for sentiment analysis where Pos. is more challenging than Neg. Thus Pos. is a better indicator of performance differences.

Transformation Based Spaces. LINEAR performs similar to C-Simple (but outperforms it for S16). This supports the hypothesis that PBC offers too little data for training mono-/bilingual embeddings. As expected BILING works better than LINEAR. Keep in mind that this is not a universal embedding space and has fewer constraints than other embedding spaces. Thus it is not directly comparable.

Unsupervised Embedding Learning. MUSE performs even worse than MONO. MAT+MPSR performs poorly, as well. This is a strong indication that PBC offers too little data for learning high-quality monolingual embeddings. In addition, we hypothesize that the word spaces themselves offer too few data points (i.e., vocabulary size) for neural network based mapping approaches. We plan to investigate this in the future. As discussed MAT+MPSR results are for a subset of 52 editions.

Context vs. Concept Based. N(t) by Dufter et al. (2018) shows consistently the best performance. Later we will see that this method works well here (massively multilingual PBC setting), but worse on EuroParl. *SAMPLE* by Dufter et al. (2018) is based on sampling like C-ID. However, it induces concepts only on a small set of pivot languages, not on all 1664 editions. This does not only work worse (C-ID beats SAMPLE except for S16), but it also requires word alignment information and is thus computationally more expensive. As has been observed frequently, S-ID is highly effective. S-ID ranks third consistently. Representing a word as its binary vector of verse occurrence provides a clear signal to the embedding learner. Concept-based methods can

be considered complementary to this feature set because they can exploit aggregated information across languages as well as across verses. C-ID alone has slightly lower performance than S-ID.

Combining Concept and Context. Co+Co, the combination, outperforms both C-ID and S-ID and seems to unite the "best of both worlds". It yields consistently the best performance with 3% to 6% relative performance increase (for μ) compared to S-ID alone and even more compared to C-ID. Overall Co+Co always ranks second. Thus combining context and concept is effective indeed, but not sufficient to outperform the strong method N(t) which is tailored for massively multilingual corpora. In experiments we found that adding C-ID to N(t) harmed the performance of N(t) severely, so these two methods seem to be incompatible. In short: N(t) is the best method on this corpus, Co+Co second-best and S-ID is third.

4 Application to a High-Resource Corpus

PBC is a low-resource, highly multilingual scenario. We now provide experimental evidence that Co+Co is broadly applicable and works in a high-resource, mildly multilingual scenario. We test the three best performing methods (based on PBC S1 μ): S-ID, Co+Co, N(t).

4.1 Data

We choose a dataset by Ammar et al. (2016), a parallel corpus covering 12 languages[6] from the proceedings of the European Parliament, Wikipedia titles and news commentary. We refer to this corpus as *EuroParl*. We do not apply any preprocessing. The dataset is lowercased.

4.2 Evaluation

Ammar et al. (2016) provide an extensive evaluation framework covering two extrinsic and four intrinsic tasks. The tasks are document classification, dependency parsing (those models were not available to us, so we omit this evaluation), word translation, word similarity, *QVEC* (Tsvetkov et al., 2015) and *QVEC-cca* (Ammar et al., 2016). For all tasks, there is a development and test set available. For more details on data and tasks see (Ammar et al., 2016). Due to obvious weaknesses of QVEC (measure not rotation invariant, for details see (Ammar et al., 2016) and

[6]Bulgarian, Czech, Danish, English, Finnish, French, German, Greek, Hungarian, Italian, Spanish, Swedish

Co+Co			S-ID		N(t)	
MinLg.	MinCount.	Word Trans.	MinCount.	Word Trans.	NPiv.	Word Trans.
6	5	18.85 $_{52.37}$				
3		18.38 $_{52.09}$	5	18.48 $_{52.37}$	**4**	**17.08** $_{59.33}$
9		20.15 $_{52.37}$	**2**	**19.22** $_{65.65}$	2	15.88 $_{50.05}$
12		19.87 $_{52.37}$	10	15.97 $_{43.92}$	6	16.90 $_{62.95}$
	2	**21.17** $_{67.41}$				
	10	17.92 $_{50.51}$				

Table 3: Hyperparameter selection on EuroParl. Initial parameter in first row; empty cell: initial parameter from first row. Bold: best result per column or selected hyperparameter values. Subscript numbers indicate coverage.

	Dim.	Word Trans.	Word Sim.	QVEC	QVEC-cca	Doc. Class.
S-ID	10	0.19	23.61	12.10	19.75	53.30
S-ID	25	4.92	42.08	**15.74**	19.24	74.33
S-ID	50	13.93	54.29	14.85	19.67	83.91
S-ID	100	15.97	53.86	11.73	25.51	86.88
S-ID	300	16.81	**56.29**	9.29	34.58	90.31
S-ID	500	**16.90**	55.51	9.01	**38.64**	**90.88**
Coverage		43.92	57.58	70.91	70.91	38.89

Table 4: Results of S-ID on the dev set for varying dimensionality. There is clear correlations with the embedding dimension. Note that results are slightly different to Table 3 due to MinCount. 10 used in this table.

Table 4) we omit QVEC in our final evaluation. In their word translation task Ammar et al. (2016) reduce the word space to contain only words from the evaluation test set. We are interested in an (unrestricted) word translation task, where all words in the word space are possible answers, which is why we reimplemented this task and report results (precision@1) only for the unrestricted word translation task. Obviously this task is more challenging and thus the performance numbers we report are significantly lower than the numbers reported by Ammar et al. (2016).

To ensure comparability with previous approaches, we follow Ammar et al. (2016) in evaluating only on words that are contained in the embedding space and simultaneously reporting the coverage (e.g., how many queries of the task are contained in the embedding space). Only for word translation we follow the same reasoning as for PBC and compute accuracy across all queries (i.e., queries not in the word space count as errors).

4.3 Hyperparameter Selection

We optimize corpus specific hyperparameters (e.g., MinLg.) on the development set of the word translation task. Co+Co: we vary the minimum number of languages that need to be covered by

	Word Trans.	Word Sim.	QVEC-cca	Doc. Class.
multiCluster*	11.79 $_{62.30}$	57.45 $_{73.89}$	43.34 $_{82.01}$	92.11 $_{48.16}$
multiCCA*	11.79 $_{77.16}$	*69.99* $_{77.94}$	41.52 $_{87.03}$	*92.18* $_{62.81}$
multiSkip*	11.70 $_{54.41}$	60.24 $_{67.55}$	36.34 $_{75.69}$	90.46 $_{45.73}$
Invariance*	12.26 $_{41.41}$	59.13 $_{62.50}$	*46.21* $_{74.78}$	91.10 $_{31.35}$
S-ID	19.13 $_{64.53}$	48.62 $_{75.39}$	24.58 $_{80.33}$	86.66 $_{56.45}$
Co+Co	***20.24*** $_{65.92}$	**52.75** $_{75.39}$	24.39 $_{82.76}$	**87.33** $_{57.52}$
N(t)	15.32 $_{59.42}$	48.01 $_{71.27}$	**25.92** $_{79.07}$	84.97 $_{53.17}$

Table 5: Results on the test set. *: methods by (Ammar et al., 2016). We downloaded their embedding spaces and performed the evaluation using their code. We can mostly reproduce their results (up to rounding errors), but word similarity numbers are slightly different. Best result across S-ID, Co+Co, N(t) is bold; best result across all methods is italic.

		Mean	Median	Stddev.	Min	Max
Bible	#editions	250	194	160	101	1530
	#tokens	259	198	172	102	2163
EuroParl	#editions	8	7	1	7	13
	#tokens	11	10	4	8	38

	#Concepts	Cov.	Cov. (relev.)	Cov. (rare)	Cov. (freq.)
Bible	119,026	0.43	0.56	0.19	0.85
EuroParl	6,208,134	0.24	0.66	0.14	0.61

Table 6: Top: Descriptive statistics of concept size. Bottom: Coverage of concepts. We report the percentage of vocabulary in English (James-Ed. for PBC) that is covered by the concepts. For "relev." we consider only words above the MinCount. threshold. To examine frequent and rare words we report the coverage on the bottom/top decile based on word frequency.

the concept identification and MinCount. for the S-ID part of Co+Co. N(t) requires pivot languages. Following Dufter et al. (2018) we choose as pivot languages those with the lowest type-token ratio (these are Greek, Danish, Spanish, French, Italian, English) and vary the number of pivot languages (*NPiv.*) between 2, 4 and 6.

Table 3 gives an overview of our hyperparameter selection. For Co+Co, we choose MinLg. 9 and MinCount. 2. For S-ID, we find the best performance with MinCount. 2. For N(t), the best result is obtained when using 4 pivot languages.

Further, we show the effect of varying embedding dimensions in Table 4. For most tasks 300 to 500 dimensions are optimal. This confirms the findings by Yin and Shen (2018). For QVEC, extremely low dimensionality is beneficial. Note that QVEC is not rotation invariant: we hypothesize that the probability of an axis being highly correlated with linguistic features in a high-dimensional space is very small compared to a low-dimensional space. QVEC-cca on the other hand benefits greatly from higher dimensions and even choosing 100 dimensions, which is a popular and reasonable choice, seriously harms the performance compared to 300 dimensions: the higher the dimensionality the more likely, CCA will find an arbitrary dimension which is highly correlated with linguistic features. When comparing our results to (Ammar et al., 2016) we need to be aware of this effect, as they used an embedding dimension of 512 vs. 100 used in this work.

4.4 Results

Table 5 provides results on the test set. It immediately becomes clear that different word spaces have different strengths and weaknesses.

Among S-ID, Co+Co and N(t), Co+Co performs best, followed by S-ID and N(t) in 3 out of 4 tasks. Only for QVEC-cca the order is different. However, the differences between methods are small in this task. Co+Co outperforms N(t) in "doc. class.", the only extrinsic task. Compared to S-ID, Co+Co yields consistent improvements (except for QVEC-cca). Co+Co provides higher coverage throughout all tasks.

The methods by (Ammar et al., 2016) perform well for "doc. class.", word similarity and QVEC-cca (the latter mostly because of increased dimensionality) and much worse for word translation. There are strong indications that neither of the four methods are applicable to highly multilingual corpora like PBC. "Invariance" considers the full cooccurrence matrix across all languages, a matrix in the size of terabytes. In addition, word alignment matrices would need to be stored. Both multiCluster and multiCCA rely on bilingual dictionaries, which are not feasible to process in the case of PBC. multiSkip requires adding $\mathcal{O}(n^2)$ terms in the objective function, which does not scale either.

In short: Among the methods that are applicable to both PBC and Europarl, Co+Co performs best, followed by S-ID and N(t).

5 Concept Quality

Table 6 reports the size of a typical concept and concept coverage with respect to the vocabulary (English). The largest concept contains 2163 tokens across 1530 editions describing the 2-gram "Simon Peter". Frequent words tend to be better covered. However, almost 20% of really rare words (i.e., hapax legomena) are contained in concepts. In Figure 3 we show five randomly sampled concepts from EuroParl.

```
hun:történelem fin:historian gre:ιστορία
spa:historia fra:histoire ita:storia eng:history

swe:heavenly swe:sword spa:heavenly spa:sword fra:heavenly
fra:sword deu:heavenly deu:sword ita:heavenly ita:sword eng:heavenly

swe:kan fin:tarkistuksen spa:enmienda fra:amendement
ita:emendamento eng:amendment

dan:erhvervet gre:οικολογική fra:filière
deu:industriezweig ita:coinvolge eng:ecological

swe:vodka fin:vodkaa gre:βότκα spa:vodka
deu:wodka ita:vodka eng:vodka
```

Figure 3: Five randomly sampled concepts from EuroParl. Quality is generally high. The second example is a video game consisting of two words.

	PBC		EuroParl					
	RTT	Sent.	Doc. Class.	Word Sim.	QVEC -cca	Word Trans.	Mean Rank	Mean Perf.
S-ID	3	3	2	2	2	2	2.33	53.91
N(t)	**1**	**1**	3	3	**1**	3	2.00	55.87
Co+Co	2	2	**1**	**1**	3	**1**	**1.67**	**56.31**

Table 7: Performance overview: we show the rank among N(t), S-ID and Co+Co across all tasks. For RTT and Sent. the overall performance is the mean over all task versions (S1, R1, S4, S16 and Pos., Neg.).

6 Related Work

We cluster prior work for **multilingual embedding learning** for parallel corpora into three groups. Our focus is on methods which are applicable to both PBC and EuroParl. **1)** follows the basic idea of projecting monolingual spaces into a unified multilingual space using (linear) transformations. We use (Mikolov et al., 2013b) together with (Duong et al., 2017) in our baseline LINEAR. Zou et al. (2013), Xiao and Guo (2014) and Faruqui and Dyer (2014) use similar approaches (e.g., by computing the transformation using CCA). It has been shown that computing the transformation using discriminator neural networks works well, even in a completely unsupervised setting. See, e.g., (Vulić and Moens, 2012; Lample et al., 2018; Chen and Cardie, 2018; Artetxe et al., 2018). We used (Lample et al., 2018) as the baseline MUSE. **2)** is true multilingual embedding learning: it integrates multilingual information in the objective of embedding learning. Klementiev et al. (2012) and Gouws et al. (2015) add a word alignment based term. Luong et al. (2015) introduce BiSkip as a bilingual extension of word2vec. For n editions, including $\mathcal{O}(n^2)$ bilingual terms does not scale. Thus this line of work is not applicable to PBC. A slightly different objective function expresses that representation of aligned sentences should be similar. Approaches based on neural networks are (Hermann and Blunsom, 2014a) (BiCVM), (Sarath Chandar et al., 2014) (autoencoders) and (Soyer et al., 2014). Again, we argue that neural network based approaches do not work for the low-resource setting of PBC. **3)** creates multilingual corpora and uses monolingual embedding learners. A successful approach is (Levy et al., 2017)'s sentence ID (S-ID). Vulić and Moens (2015) create pseudocorpora by merging words from multiple languages into a single corpus. Dufter et al. (2018) found this method to perform poorly on PBC. Søgaard et al. (2015) learn a space by factorizing an interlingual matrix based on Wikipedia concepts. word2vec is roughly equivalent to matrix factorization (Levy and Goldberg, 2014), so this work fits this group.

With the raise of pretrained language models, methods to obtain **multilingual contextual representations** have been proposed (Conneau and Lample, 2019). We focus on creating static word embedings which are computationally much more efficient at the cost of lower performance.

Much research has been dedicated to identifying **multilingual concepts**. BabelNet (Navigli and Ponzetto, 2012) leverages existing resources (mostly manual annotations), including Wikipedia, using information extraction methods. BabelNet could be used to learn concept based embeddings, but it covers only 284 languages and thus cannot be applied to all PBC languages. Other work induces concepts within a dictionary graph (Ammar et al., 2016; Dufter et al., 2018), with alignment algorithms (Östling, 2014), or by means of sampling (Lardilleux and Lepage, 2009). We used sampling based concept induction in this work, as it scales easily for 1000+ languages.

7 Summary

We proposed Co+Co, to the best of our knowledge the first method that learns embeddings *jointly from concept and context information*. We showed that Co+Co performs well across two very different corpora and a wide range of tasks. Among the three high-performing methods applicable to both PBC and EuroParl Co+Co performs best (see Table 7). Two other advantages of Co+Co are that it is a simple method (compared to more complex methods like MUSE) and scalable to 1000s of languages. In summary, Co+Co is a simple, strong and scalable method that is well suited for a wide range of application scenarios.

We gratefully **acknowledge** funding for this

work by the European Research Council (ERC #740516) and by Zentrum Digitalisierung.Bayern (ZD.B), the digital technology initiative of the State of Bavaria.

References

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

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.

Xilun Chen and Claire Cardie. 2018. Unsupervised multilingual word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067.

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*.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2017. Multilingual training of crosslingual word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Karl Moritz Hermann and Phil Blunsom. 2014a. Multilingual distributed representations without word alignment. In *Proceedings of the 2014 International Conference on Learning Representations*.

Karl Moritz Hermann and Phil Blunsom. 2014b. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of the 24th International Conference on Computational Linguistics*.

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

Adrien Lardilleux and Yves Lepage. 2009. Sampling-based multilingual alignment. In *Proceedings of 7th Conference on Recent Advances in Natural Language Processing*.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*.

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*.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel bible corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*.

Ryan T. McDonald, Slav Petrov, and Keith B. Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

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

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

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

Robert Östling. 2014. Bayesian word alignment for massively parallel texts. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Roger Penrose. 1956. On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

AP Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh M. Khapra, Balaraman Ravindran, Vikas C. Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In *Proceedings of the 2014 Annual Conference on Neural Information Processing Systems*.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual nlp. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing*.

Hubert Soyer, Pontus Stenetorp, and Akiko Aizawa. 2014. Leveraging monolingual data for crosslingual compositional word representations. In *Proceedings of the 2015 International Conference on Learning Representations*.

Morris Swadesh. 1946. South Greenlandic (Eskimo). In Cornelius Osgood, editor, *Linguistic Structures of Native America*. Viking Fund Inc. (Johnson Reprint Corp.), New York.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Ivan Vulić and Marie-Francine Moens. 2012. Detecting highly confident word translations from comparable corpora without any prior knowledge. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*.

Ivan Vulić 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 and the 7th International Joint Conference on Natural Language Processing*.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the 18th Conference on Computational Natural Language Learning*.

Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding. In *Advances in Neural Information Processing Systems*.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*.

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*.

Chapter 4

Analytical Methods for Interpretable Ultradense Word Embeddings

Analytical Methods for Interpretable Ultradense Word Embeddings

Philipp Dufter, Hinrich Schütze
Center for Information and Language Processing (CIS)
LMU Munich, Germany
philipp@cis.lmu.de

Abstract

Word embeddings are useful for a wide variety of tasks, but they lack interpretability. By rotating word spaces, interpretable dimensions can be identified while preserving the information contained in the embeddings without any loss. In this work, we investigate three methods for making word spaces interpretable by rotation: Densifier (Rothe et al., 2016), linear SVMs and DensRay, a new method we propose. In contrast to Densifier, DensRay can be computed in closed form, is hyperparameter-free and thus more robust than Densifier. We evaluate the three methods on lexicon induction and set-based word analogy. In addition we provide qualitative insights as to how interpretable word spaces can be used for removing gender bias from embeddings.

1 Introduction

Distributed representations for words have been of interest in natural language processing for many years. Word embeddings have been particularly effective and successful. On the downside, embeddings are generally not interpretable. But interpretability is desirable for several reasons. i) Semantically or syntactically similar words can be extracted: e.g., for lexicon induction. ii) Interpretable dimensions can be used to evaluate word spaces by examining which information is covered by the embeddings. iii) Computational advantage: for a high-quality sentiment classifier only a couple of dimensions of a high-dimensional word space are relevant. iv) By removing interpretable dimensions one can remove unwanted information (e.g., gender bias). v) Most importantly, interpretable embeddings support the goal of interpretable deep learning models.

Orthogonal transformations have been of particular interest in the literature. The reason is twofold: under the assumption that existing word embeddings are of high-quality one would like to preserve the original embedding structure by using orthogonal transformations (i.e., preserving original distances). Park et al. (2017) provide evidence that rotating existing dense word embeddings achieves the best performance across a range of interpretability tasks.

In this work we modify the objective function of Densifier (Rothe et al., 2016) such that a closed form solution becomes available. We call this method DensRay. Following Amir et al. (2015) we compute simple linear SVMs, which we find to perform surprisingly well. We compare these methods on the task of lexicon induction.

Further, we show how interpretable word spaces can be applied to other tasks: first we use interpretable word spaces for debiasing embeddings. Second we show how they can be used for solving the set-based word analogy task. To this end, we introduce the set-based method IntCos, which is closely related to LRCos introduced by Drozd et al. (2016). We find IntCos to perform comparable to LRCos, but to be preferable for analogies which are hard to solve.

Our contributions are: **i)** We modify Densifier's objective function and derive an analytical solution for computing interpretable embeddings. **ii)** We show that the analytical solution performs as well as Densifier but is more robust. **iii)** We provide evidence that simple linear SVMs are best suited for the task of lexicon induction. **iv)** We demonstrate how interpretable embedding spaces can be used for debiasing embeddings and solving the set-based word analogy task. The source code of our experiments is available.[1]

[1] https://github.com/pdufter/densray

67

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1185–1191,
Hong Kong, China, November 3–7, 2019.

2 Methods

2.1 Notation

We consider a vocabulary $V := \{v_1, v_2, ..., v_n\}$ together with an embedding matrix $E \in \mathbb{R}^{n \times d}$ where d is the embedding dimension. The ith row of E is the vector e_i.[2] We require an annotation for a specific linguistic feature (e.g., sentiment) and denote this annotation by $l : V \to \{-1, 1\}$. The objective is to find an orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ such that EQ is interpretable, i.e., the values of the first k dimensions correlate well with the linguistic feature. We refer to the first k dimensions as interpretable ultradense word space. We interpret $x \in \mathbb{R}^n$ as a column vector and $x^\intercal$ as a row vector. Further, we normalize all word embeddings with respect to the euclidean norm.

2.2 DensRay

Throughout this section $k = 1$. Given a linguistic signal l (e.g., sentiment), consider $L_= := \{(v, w) \in V \times V |\ l(v) = l(w)\}$, and analogously $L_{\neq}$. We call $d_{vw} := e_v - e_w$ a difference vector.

Densifier (Rothe et al., 2016) solves the following optimization problem,

$$\max_q \sum_{(v,w) \in L_{\neq}} \alpha_{\neq} \left\| q^\intercal d_{vw} \right\|_2 - \sum_{(v,w) \in L_=} \alpha_= \left\| q^\intercal d_{vw} \right\|_2 ,$$

subject to $q^\intercal q = 1$ and $q \in \mathbb{R}^d$. Further $\alpha_{\neq}, \alpha_= \in [0, 1]$ are hyperparameters. We now modify the objective function: we use the squared euclidean norm instead of the euclidean norm, something that is frequently done in optimization to simplify the gradient. The problem becomes then

$$\max_q \sum_{(v,w) \in L_{\neq}} \alpha_{\neq} \left\| q^\intercal d_{vw} \right\|_2^2 - \sum_{(v,w) \in L_=} \alpha_= \left\| q^\intercal d_{vw} \right\|_2^2 . \quad (1)$$

Using $\|x\|_2^2 = x^\intercal x$ together with associativity of the matrix product we can simplify to

$$\max_q q^\intercal \Big(\alpha_{\neq} \sum_{(v,w) \in L_{\neq}} d_{vw} d_{vw}^\intercal - \alpha_= \sum_{(v,w) \in L_=} d_{vw} d_{vw}^\intercal \Big) q \quad (2)$$
$$=: \max_q q^\intercal A q \quad \text{subject to } q^\intercal q = 1.$$

[2] We denote the vector corresponding to a word w by e_w.

Thus we aim to maximize the Rayleigh quotient of A and q. Note that A is a real symmetric matrix. Then it is well known that the eigenvector belonging to the maximal eigenvalue of A solves the above problem (cf. Horn et al. (1990, Section 4.2)). We call this analytical solution **DensRay**.

A second dimension that is orthogonal to the first dimension and encodes the linguistic features second strongest is given by the eigenvector corresponding to the second largest eigenvalue. The matrix of k eigenvectors of A ordered by the corresponding eigenvalues yields the desired matrix Q (cf. Horn et al. (1990, Section 4.2)) for $k > 1$. Due to A being a real symmetric matrix, Q is always orthogonal.

2.3 Comparison to Densifier

We have shown that DensRay is a closed form solution to our new formalization of Densifier. This formalization entails differences.

Case $k > 1$. While both methods – Densifier and DensRay – yield ultradense k dimensional subspaces. While we show that the spaces are comparable for $k = 1$ we leave it to future work to examine how the subspaces differ for $k > 1$.

Multiple linguistic signals. Given multiple linguistic features, Densifier can obtain a single orthogonal transformation simultaneously for all linguistic features with chosen dimensions reserved for different features. DensRay can encode multiple linguistic features in one transformation only by iterative application.

Optimization. Densifier is based on solving an optimization problem using stochastic gradient descent with iterative orthogonalization of Q. DensRay, in contrast, is an analytical solution. Thus we expect DensRay to be more robust, which is confirmed by our experiments.

2.4 Geometric Interpretation

Assuming we normalize the vectors d_{vw} one can interpret Eq. 1 as follows: we search for a unit vector q such that the square of the cosine similarity with d_{vw} is large if $(v, w) \in L_{\neq}$ and small if $(v, w) \in L_=$. Thus, we identify dimensions that are parallel/orthogonal to difference vectors of words belonging to different/same classes. It seems reasonable to consider the average cosine similarity. Thus if $n_=$, $n_{\neq}$ is the number of elements in $L_=$, $L_{\neq}$ one can choose $\alpha_{\neq} = n_{\neq}^{-1}$ and $\alpha_= = n_=^{-1}$.

3 Lexicon Induction

We show that DensRay and Densifier indeed perform comparably using the task of lexicon induction. We adopt Rothe et al. (2016)'s experimental setup. We also use Rothe et al. (2016)'s code for Densifier. Given a word embedding space and a sentiment/concreteness dictionary (binary or continuous scores where we binarize continuous scores using the median), we identify a one-dimensional interpretable subspace. Subsequently we use the values along this dimension to predict a score for unseen words and report Kendall's τ rank correlation with the gold scores.

To ensure comparability across methods we have redone all experiments in the same setting: we deduplicated lexicons, removed a potential train/test overlap and ignored neutral words in the lexicons. We set $\alpha_{\neq} = \alpha_{=} = 0.5$ to ensure comparability between Densifier and DensRay.

Additionally we report results created by linear SVM/SVR inspired be their good performance as demonstrated by Amir et al. (2015). While they did not use linear kernels, we require linear kernels to obtain interpretable dimensions. Naturally the normal vector of the hyperplane in SVMs/SVRs reflects an interpretable dimension. An orthogonal transformation can be computed by considering a random orthogonal basis of the null space of the interpretable dimension.

Table 1 shows results. As expected the performance of Densifier and DensRay is comparable (macro mean deviation of 0.001). We explain slight deviations between the results with the slightly different objective functions of DensRay and Densifier. In addition, the re-orthogonalization used in Densifier can result in an unstable training process. Figure 1 assesses the stability by reporting mean and standard deviation for the concreteness task (BWK lexicon). We varied the size of the training lexicon as depicted on the x-axis and sampled 40 subsets of the lexicon with the prescribed size. For the sizes 512 and 2048 Densifier shows an increased standard deviation. This is because there is at least one sample for which the performance significantly drops. Removing the re-orthogonalization in Densifier prevents the drop and restores performance. Recent work (Zhao and Schütze, 2019) also finds that replacing the orthogonalization with a regularization is reasonable in certain circumstances. Given that DensRay and Densifier yield the same performance and DensRay is a stable closed form solution always yielding a orthogonal transformation we conclude that DensRay is preferable.

Surprisingly, simple linear SVMs perform best in the task of lexicon induction. SVR is slightly better when continuous lexica are used for training (line 8). Note that the eigendecomposition used in DensRay yields a basis with dimensions ordered by their correlation with the linguistic feature. An SVM can achieve this only by iterated application.

	Task	Emb.	Lex. (Train)	Lex. (Test)	Dens.	DensRay	SVR	SVM
1	sent	CZ	SubLex	SubLex	0.546	0.549	**0.585**	0.585
2	sent	DE	GermanPC	GermanPC	0.636	0.631	0.674	**0.677**
3	sent	ES	fullstrength	fullstrength	0.541	0.546	0.571	**0.576**
4	sent	FR	FEEL	FEEL	0.469	0.471	0.555	**0.565**
5	sent	EN	WHM	WHM	0.623	0.623	**0.627**	0.625
6	sent	EN(t)	WHM	SE Trial*	0.624	0.621	0.618	**0.637**
7	sent	EN(t)	WHM	SE Test*	0.600	0.608	0.619	**0.636**
8	conc	EN	BWK*	BWK*	0.599	0.602	**0.655**	0.641
9			Macro Mean		0.580	0.581	0.613	**0.618**

Table 1: Results on lexicon induction. Numbers are Kendall τ rank correlation. For details on the resources see Table 2 and (Rothe et al., 2016). Bold: best result across methods. *: continuous lexicon.

Name	Description
CZ, DE, ES	Czech, German, Spanish embeddings by (Rothe et al., 2016)
FR	French frWac embeddings (Fauconnier, 2015)
EN	English GoogleNews embeddings (Mikolov et al., 2013)
EN(t)	English Twitter Embeddings (Rothe et al., 2016)

Name	Description
SubLex	Czech sentiment lexicon (Veselovská and Bojar, 2013)
GermanPC	German sentiment lexicon (Waltinger, 2010)
fullstrength	Spanish sentiment lexicon (Perez-Rosas et al., 2012)
FEEL	French sentiment lexicon (Abdaoui et al., 2017)
WHM	English sentiment lexicon; combination of MPQA (Wilson et al., 2005), Opinion Lexicon (Hu and Liu, 2004) and NRC emotion lexcion (Mohammad and Turney, 2013)
SE	Semeval 2015 Task 10E shared task data (Rosenthal et al., 2015)
BWK	English concreteness lexicon (Brysbaert et al., 2014)

Table 2: Overview of resources for lexicon induction. The setup is identical to (Rothe et al., 2016).

4 Removing Gender Bias

Word embeddings are well-known for encoding prevalent biases and stereotypes (cf. Bolukbasi et al. (2016)). We demonstrate qualitatively that by identifying an interpretable gender dimension and subsequently removing this dimension, one can remove parts of gender information that potentially could cause biases in downstream processing. Given the original word space E we consider the interpretable space $E' := EQ$, where Q is computed using DensRay. We denote by $E_{\cdot,-1} \in \mathbb{R}^{n\times(d-1)}$ the word space with removed first dimension and call it the "complement" space. We expect $E_{\cdot,-1}$ to be a word space with less gender bias.

To examine this approach qualitatively we use

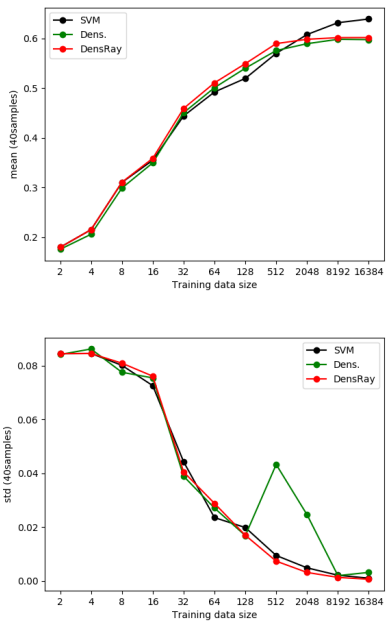


Figure 1: Mean (top) and standard deviation (bottom) of the performance across 40 samples of the training lexicon with varying sizes. Performed on the English concreteness task (line 8 in Table 1). SVR performs similar to SVM and is omitted for clarity.

a list of occupation names[3] by Bolukbasi et al. (2016) and examine the cosine similarities of occupations with the vectors of "man" and "woman". Figure 2 shows the similarities in the original space E and debiased space $E_{\cdot,-1}$. One can see the similarities are closer to the identity (i.e., same distance to "man" and "woman") in the complement space. To identify occupations with the greatest bias, Table 3 lists occupations for which $\text{sim}(e_w, e_{\text{man}}) - \text{sim}(e_w, e_{\text{woman}})$ is largest/smallest. One can clearly see a debiasing effect when considering the complement space. Extending this qualitative study to a more rigorous quantitative evaluation is part of future work.

	Original Space			Complement Space		
		man	woman		man	woman
female bias	actress	0.23	0.46	lawyer	0.16	0.27
	businesswoman	0.32	0.53	ambassador	0.07	0.17
	registered_nurse	0.12	0.33	attorney	0.05	0.15
	housewife	0.34	0.55	legislator	0.26	0.36
	homemaker	0.22	0.40	minister	0.10	0.20
	...					
male bias	hitman	0.41	0.27	captain	0.31	0.24
	gangster	0.34	0.20	marksman	0.29	0.21
	skipper	0.27	0.11	maestro	0.28	0.20
	marksman	0.31	0.14	hitman	0.40	0.32
	maestro	0.30	0.12	skipper	0.25	0.17

Table 3: Top 5 occupations that exhibit the greatest bias (measured by difference in cosine similarity). Numbers indicate cosine similarity between word vectors.

[3] https://github.com/tolga-b/debiaswe/blob/master/data/professions.json

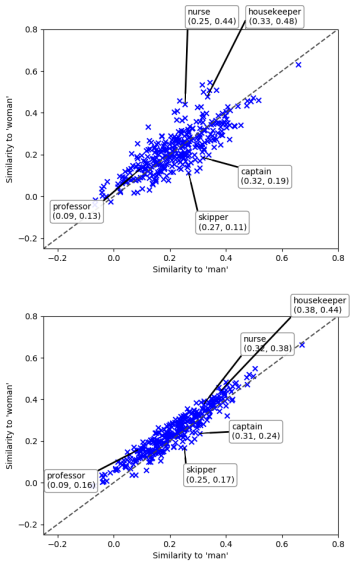


Figure 2: Similarities of occupation vectors with the vectors of man and woman. Top shows the original word space and bottom the word space with removed gender dimension.

5 Word Analogy

In this section we use interpretable word spaces for set-based word analogy. Given a list of analogy pairs $[(a, a'), (b, b'), (c, c'), \ldots]$ the task is to predict a' given a. Drozd et al. (2016) provide a detailed overview over different methods, and find that their method LRCos performs best.

LRCos assumes two classes: all left elements of a pair ("left class") and all right elements ("right class"). They train a logistic regression (LR) to differentiate between these two classes. The predicted score of the LR multiplied by the cosine similarity in the word space is their final score. Their prediction for a' is the word with the highest final score.

We train the classifier on all analogy pairs except for a single pair for which we then obtain the predicted score. In addition we ensure that no word belonging to the test analogy is used during training (splitting the data only on word analogy pairs is not sufficient).

Inspired by LRCos we use interpretable word spaces for approaching word analogy: we train DensRay or an SVM to obtain interpretable embeddings $E' = EQ$ using the class information as reasoned above. We use a slightly different notation in this section: for a word w the ith component of its embedding is given by $E_{w,i}$. Therefore we denote as $E_{\cdot,1}$ the first column of E' (i.e., the

most interpretable dimension). We min-max normalize $E_{\cdot,1}$ such that words belonging to the right class have a high value (i.e., we flip the sign if necessary). For a query word a we now want to identify the corresponding a' by solving

$$\hat{a} = \arg\max_{v \in V} \text{norm}(E_{v,1}) \, \text{sim}(E_{a,\cdot}, E_{v,\cdot})$$

where *sim* computes the cosine similarity.

Given the result from §4 we extend the above method by computing the cosine similarity in the orthogonal complement, i.e., $\text{sim}(E_{a,-1}, E_{v,-1})$. We call this method **IntCos** (INTerpretable, COSine). Depending on the space used for computing the cosine similarity add the word "Original" or "Complement".

We evaluate this method across two analogy datasets. These are the Google Analogy Dataset (GA) (Mikolov et al., 2013) and BATS (Drozd et al., 2016). As embeddings spaces we use Google News Embeddings (GN) (Mikolov et al., 2013) and FastText subword embeddings (FT) (Bojanowski et al., 2017). We consider the first 80k word embeddings from each space.

Table 4 shows the results. The first observation is that there is no clear winner. IntCos Original performs comparably to LRCos with slight improvements for GN/BATS: here the classes are widespread and exhibit low cosine similarity (IntraR and IntraL), which makes them harder to solve. IntCos Complement maintains performance for GN/BATS and is beneficial for Derivational analogies on GN. For most other analogies it harms performance.

Within IntCos Original it is favorable to use DensRay as it gives slight performance improvements. Especially for harder analogies, where interclass similarity is high and intraclass similarities are low (e.g., in GN/BATS), DensRay outperforms SVMs. In contrast to SVMs, DensRay considers difference vectors *within* classes as well – this seems to be of advantage here.

6 Related Work

Identifying Interpretable Dimensions. Most relevant to our method is a line of work that uses transformations of existing word spaces to obtain interpretable subspaces. Rothe et al. (2016) compute an orthogonal transformation using shallow neural networks. Park et al. (2017) apply exploratory factor analysis to embedding spaces

		Mean Cosine Sim			Precision				
					IntCos				LRCos
					complement		original		
		Inter	IntraL	IntraR	DensR.	SVM	DensR.	SVM	
FT/BATS	Inflectional	0.75	0.48	0.51	0.92	0.93	**0.97**	**0.97**	**0.97**
	Derivational	0.63	0.47	0.45	0.74	0.78	**0.81**	0.80	0.80
	Encyclopedia	0.48	0.43	0.55	0.30	0.43	0.41	0.43	**0.45**
	Lexicography	0.62	0.37	0.38	0.17	0.20	0.21	0.22	**0.26**
	Macro Mean	0.62	0.44	0.47	0.53	0.58	0.60	0.60	**0.61**
	Macro Std	0.12	0.06	0.09	0.34	0.33	0.34	0.33	**0.32**
GN/BATS	Inflectional	0.63	0.22	0.23	**0.88**	0.87	**0.88**	**0.88**	**0.88**
	Derivational	0.44	0.21	0.20	**0.55**	0.50	0.51	0.48	0.44
	Encyclopedia	0.35	0.29	0.42	0.33	**0.35**	**0.35**	0.32	0.34
	Lexicography	0.45	0.17	0.18	**0.19**	0.17	**0.19**	0.17	0.18
	Macro Mean	0.46	0.22	0.26	**0.48**	0.47	**0.48**	0.46	0.45
	Macro Std	0.14	0.07	0.12	**0.31**	**0.31**	0.32	0.32	0.32
FT/GA	Micro Mean	0.73	0.48	0.53	0.88	0.91	**0.93**	0.92	**0.93**
	Macro Mean	0.71	0.50	0.53	0.87	0.90	**0.91**	0.90	0.89
	Macro Std	0.11	0.05	0.06	0.11	**0.08**	0.12	0.17	0.23
GN/GA	Micro Mean	0.62	0.31	0.36	0.85	0.87	**0.89**	0.87	0.88
	Macro Mean	0.61	0.30	0.35	0.85	0.86	**0.88**	0.85	0.87
	Macro Std	0.10	0.09	0.10	0.08	**0.07**	0.09	0.11	0.11

Table 4: Left part shows mean cosine similarity. Inter: mean cosine similarity between pairs. IntraL/R: mean cosine similarity within the left/right class. Right part shows precision for word analogy task.

to obtain interpretable dimensions in an unsupervised manner. Their approach relies on solving complex optimization problems, while we focus on closed form solutions. Senel et al. (2018) use SEMCAT categories in combination with the Bhattacharya distance to identify interpretable directions. Also, oriented PCA (Diamantaras and Kung, 1996) is closely related to our method. However, both methods yield non-orthogonal transformation. Faruqui et al. (2015a) use semantic lexicons to retrofit embedding spaces. Thus they do not fully maintain the structure of the word space, which is in contrast to this work.

Interpretable Embedding Algorithms. Another line of work modifies embedding algorithms to yield interpretable dimensions (Koç et al., 2018; Luo et al., 2015; Shin et al., 2018; Zhao et al., 2018). There is also much work that generates sparse embeddings that are claimed to be more interpretable (Murphy et al., 2012; Faruqui et al., 2015b; Fyshe et al., 2015; Subramanian et al., 2018). Instead of learning new embeddings, we aim at making dense embeddings interpretable.

7 Conclusion

We investigated analytical methods for obtaining interpretable word embedding spaces. Relevant methods were examined with the tasks of lexicon induction, word analogy and debiasing.

We gratefully **acknowledge** funding through a Zentrum Digitalisierung.Bayern fellowship awarded to the first author. This work was supported by the European Research Council (# 740516).

References

Amine Abdaoui, Jérôme Azé, Sandra Bringay, and Pascal Poncelet. 2017. Feel: a french expanded emotion lexicon. *Language Resources and Evaluation*, 51(3).

Silvio Amir, Ramón Astudillo, Wang Ling, Bruno Martins, Mario J Silva, and Isabel Trancoso. 2015. Inesc-id: A regression model for large scale twitter sentiment lexicon induction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*.

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

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3).

Konstantinos I Diamantaras and Sun Yuan Kung. 1996. *Principal component neural networks: theory and applications*, volume 5. Wiley New York.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015a. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A Smith. 2015b. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Jean-Philippe Fauconnier. 2015. French word embeddings.

Alona Fyshe, Leila Wehbe, Partha P Talukdar, Brian Murphy, and Tom M Mitchell. 2015. A compositional and interpretable semantic space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Roger A Horn, Roger A Horn, and Charles R Johnson. 1990. *Matrix analysis*. Cambridge university press.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Aykut Koç, Ihsan Utlu, Lutfi Kerem Senel, and Haldun M Ozaktas. 2018. Imparting interpretability to word embeddings. *arXiv preprint arXiv:1807.07279*.

Hongyin Luo, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2015. Online learning of interpretable word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

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

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3).

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. *Proceedings of the 24th International Conference on Computational Linguistics*.

Sungjoon Park, JinYeong Bak, and Alice Oh. 2017. Rotated word vector representations and their interpretability. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Veronica Perez-Rosas, Carmen Banea, and Rada Mihalcea. 2012. Learning sentiment lexicons in spanish. In *Proceedings of the seventh international conference on Language Resources and Evaluation*.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th international workshop on semantic evaluation*.

Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. 2016. Ultradense word embeddings by orthogonal transformation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Lutfi Kerem Senel, Ihsan Utlu, Veysel Yucesoy, Aykut Koc, and Tolga Cukur. 2018. Semantic structure and interpretability of word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Jamin Shin, Andrea Madotto, and Pascale Fung. 2018. Interpreting word embeddings with eigenvector analysis. *openreview.net*.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. 2018. Spine: Sparse interpretable neural embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kateřina Veselovská and Ondřej Bojar. 2013. Czech sublex 1.0. Charles University, Faculty of Mathematics and Physics.

Ulli Waltinger. 2010. Germanpolarityclues: A lexical resource for german sentiment analysis. In *Proceedings of the seventh international conference on Language Resources and Evaluation*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang Chang. 2018. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Mengjie Zhao and Hinrich Schütze. 2019. A multilingual bpe embedding space for universal sentiment lexicon induction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*.

Supplementary Material to "Analytical Methods for Interpretable Ultradense Word Embeddings"

Philipp Dufter, Hinrich Schütze
Center for Information and Language Processing (CIS)
LMU Munich, Germany
philipp@cis.lmu.de

1 Code

The code which was used to conduct the experiments in this paper is available at https://github.com/pdufter/densray.

2 Continuous Lexicon

In case of a continuous lexicon $l : V \to \mathbb{R}$ one can extend Equation 2 in the main paper by defining:

$$A := \sum_{(v,w) \in V \times V} -l(v)l(w)d_{vw}d_{vw}^{\mathsf{T}}$$

In the case of a binary lexicon Equation 2 from the main paper is recovered for $\alpha_{\neq} = \alpha_{=} = 1$.

3 Full Analogy Results

In this section we present the results of the word analogy task per category. See Table 1 and Table 2 for detailed results with the methods IntCos Complement and Original, respectively. The format and numbers presented are the same as in the corresponding table from the main paper.

FastText

Google Analogy

	Mean Cosine Sim			Precision		
				IntCos		LRCos
	InterIntraL		IntraR	DensRay	SVM	
capital-common-countries	0.76	0.53	0.56	**1.00**	**1.00**	**1.00**
capital-world	0.75	0.44	0.51	0.97	0.96	**1.00**
city-in-state	0.71	0.51	0.63	0.78	0.79	**0.85**
currency	0.33	0.59	0.48	0.62	**0.69**	0.08
family	0.84	0.57	0.59	0.91	0.91	**0.95**
gram1-adjective-to-adverb	0.66	0.50	0.56	0.78	0.84	**0.88**
gram2-opposite	0.72	0.51	0.54	0.69	**0.83**	0.76
gram3-comparative	0.75	0.53	0.57	0.84	0.89	**0.95**
gram4-superlative	0.70	0.53	0.61	0.94	**1.00**	**1.00**
gram5-present-participle	0.76	0.43	0.48	**1.00**	0.94	**1.00**
gram6-nationality-adjective	0.70	0.55	0.54	0.90	0.90	**0.93**
gram7-past-tense	0.73	0.49	0.47	0.82	0.90	**0.97**
gram8-plural	0.80	0.41	0.42	**1.00**	**1.00**	**1.00**
gram9-plural-verbs	0.74	0.43	0.48	0.90	0.97	**1.00**
Micro Mean	0.73	0.48	0.53	0.88	0.91	**0.93**
Macro Mean	0.71	0.50	0.53	0.87	**0.90**	0.89
Macro Std	0.11	0.05	0.06	0.11	0.08	**0.23**

Google News

Google Analogy

	Mean Cosine Sim			Precision		
				IntCos		LRCos
	InterIntraL		IntraR	DensRay	SVM	
capital-common-countries	0.64	0.37	0.38	0.91	**0.96**	0.91
capital-world	0.64	0.34	0.36	0.86	0.88	**0.90**
city-in-state	0.59	0.37	0.49	0.82	0.85	**0.87**
currency	0.37	0.42	0.44	**0.78**	0.72	0.56
family	0.74	0.48	0.53	0.83	**0.87**	**0.87**
gram1-adjective-to-adverb	0.49	0.21	0.26	0.69	**0.75**	**0.75**
gram2-opposite	0.50	0.24	0.31	0.68	**0.75**	0.71
gram3-comparative	0.60	0.25	0.40	0.92	0.92	**0.97**
gram4-superlative	0.54	0.26	0.39	**0.97**	0.91	0.94
gram5-present-participle	0.70	0.20	0.21	0.88	0.88	**0.94**
gram6-nationality-adjective	0.72	0.41	0.41	**0.95**	**0.95**	0.93
gram7-past-tense	0.66	0.21	0.22	0.82	0.82	**0.85**
gram8-plural	0.73	0.21	0.23	0.92	0.89	**0.97**
gram9-plural-verbs	0.64	0.22	0.24	0.87	0.90	**0.93**
Micro Mean	0.62	0.31	0.36	0.85	0.87	**0.88**
Macro Mean	0.61	0.30	0.35	0.85	0.86	**0.87**
Macro Std	0.10	0.09	0.10	0.08	0.07	**0.11**

BATS

	Mean Cosine Sim			Precision		
				IntCos		LRCos
	InterIntraL		IntraR	DensRay	SVM	
Derivational	0.63	0.47	0.45	0.74	0.78	**0.80**
D01 [noun+less-reg]	0.51	0.36	0.45	0.50	0.53	**0.60**
D02 [un+adj-reg]	0.71	0.46	0.46	0.68	0.72	**0.84**
D03 [adj+ly-reg]	0.63	0.50	0.52	0.86	0.88	**0.90**
D04 [over+adj-reg]	0.63	0.45	0.45	0.59	**0.69**	0.62
D05 [adj+ness-reg]	0.63	0.49	0.51	0.92	**1.00**	0.92
D06 [re+verb-reg]	0.74	0.52	0.49	0.50	0.62	**0.76**
D07 [verb+able-reg]	0.57	0.48	0.45	**0.71**	0.66	0.63
D08 [verb+er-irreg]	0.55	0.48	0.41	0.84	**0.88**	0.79
D09 [verb+tion-irreg]	0.63	0.46	0.41	0.86	**0.88**	0.86
D10 [verb+ment-irreg]	0.64	0.46	0.42	0.86	0.88	**0.90**
Encyclopedia	0.48	0.43	0.55	0.30	0.43	**0.45**
E01 [country - capital]	0.63	0.47	0.41	0.72	0.96	**0.98**
E02 [country - language]	0.40	0.43	0.59	0.24	**0.35**	0.33
E03 [UK-city - county]	0.59	0.49	0.57	0.22	**0.36**	**0.36**
E04 [name - nationality]	0.28	0.39	0.64	0.45	**0.66**	0.60
E05 [name - occupation]	0.44	0.41	0.57	0.42	0.65	**0.73**
E06 [animal - young]	0.47	0.43	0.44	0.05	0.07	**0.15**
E07 [animal - sound]	0.37	0.43	0.40	0.15	0.20	**0.22**
E08 [animal - shelter]	0.44	0.42	0.51	0.00	0.07	**0.13**
E09 [things - color]	0.44	0.38	0.81	0.04	**0.22**	0.16
E10 [male - female]	0.73	0.43	0.43	0.68	0.68	**0.78**
Inflectional	0.75	0.48	0.51	0.92	0.93	**0.97**
I01 [noun - plural-reg]	0.79	0.39	0.41	0.98	**1.00**	**1.00**
I02 [noun - plural-irreg]	0.77	0.40	0.42	0.80	0.80	**0.84**
I03 [adj - comparative]	0.75	0.50	0.52	0.97	**1.00**	**1.00**
I04 [adj - superlative]	0.71	0.51	0.58	0.96	0.96	**1.00**
I05 [verb-inf - 3pSg]	0.77	0.52	0.53	0.96	0.98	**1.00**
I06 [verb-inf - Ving]	0.77	0.51	0.51	0.80	0.88	**0.96**
I07 [verb-inf - Ved]	0.75	0.51	0.55	0.92	0.94	**1.00**
I08 [verb-Ving - 3pSg]	0.70	0.48	0.51	0.94	0.96	**0.98**
I09 [verb-Ving - Ved]	0.73	0.50	0.54	0.92	0.88	**0.98**
I10 [verb-3pSg - Ved]	0.72	0.53	0.55	0.96	0.96	**1.00**
Lexicography	0.62	0.37	0.38	0.17	0.20	**0.26**
L01 [hypernyms - animals]	0.58	0.43	0.53	0.02	**0.29**	0.24
L02 [hypernyms - misc]	0.55	0.35	0.39	**0.14**	0.11	**0.14**
L03 [hyponyms - misc]	0.63	0.35	0.31	**0.28**	**0.28**	**0.28**
L04 [meronyms - substance]	0.53	0.36	0.44	0.15	**0.21**	0.17
L05 [meronyms - member]	0.58	0.36	0.36	0.10	0.10	**0.12**
L06 [meronyms - part]	0.53	0.31	0.30	0.04	**0.09**	**0.09**
L07 [synonyms - intensity]	0.67	0.37	0.37	0.25	0.25	**0.36**
L08 [synonyms - exact]	0.71	0.33	0.31	0.18	0.16	**0.22**
L09 [antonyms - gradable]	0.68	0.45	0.43	0.35	0.33	**0.55**
L10 [antonyms - binary]	0.72	0.40	0.40	0.18	0.18	**0.39**
Micro Mean	0.62	0.44	0.47	0.52	0.58	**0.61**
Macro Mean	0.62	0.44	0.47	0.53	0.58	**0.61**
Macro Std	0.12	0.06	0.09	**0.34**	0.33	0.32

BATS

	Mean Cosine Sim			Precision		
				IntCos		LRCos
	InterIntraL		IntraR	DensRay	SVM	
Derivational	0.44	0.21	0.20	**0.55**	0.50	0.44
D01 [noun+less-reg]	0.26	0.16	0.24	**0.14**	**0.14**	0.05
D02 [un+adj-reg]	0.47	0.17	0.20	**0.66**	0.54	0.58
D03 [adj+ly-reg]	0.48	0.17	0.22	0.70	**0.76**	**0.76**
D04 [over+adj-reg]	0.39	0.17	0.21	0.41	**0.44**	0.30
D05 [adj+ness-reg]	0.47	0.21	0.26	**0.75**	**0.75**	0.65
D06 [re+verb-reg]	0.56	0.29	0.28	**0.64**	0.53	0.58
D07 [verb+able-reg]	0.38	0.25	0.20	**0.38**	0.28	0.19
D08 [verb+er-irreg]	0.30	0.24	0.17	**0.29**	0.19	0.07
D09 [verb+tion-irreg]	0.51	0.22	0.15	**0.73**	0.63	0.51
D10 [verb+ment-irreg]	0.47	0.24	0.15	**0.60**	0.56	0.44
Encyclopedia	0.35	0.29	0.42	0.33	**0.35**	0.34
E01 [country - capital]	0.61	0.35	0.32	0.88	**0.90**	**0.90**
E02 [country - language]	0.36	0.31	0.45	**0.47**	0.30	0.36
E03 [UK-city - county]	0.41	0.36	0.52	**0.14**	**0.14**	**0.14**
E04 [name - nationality]	0.20	0.20	0.39	**0.33**	**0.33**	0.26
E05 [name - occupation]	0.33	0.21	0.40	0.45	**0.62**	0.52
E06 [animal - young]	0.34	0.36	0.38	0.06	0.06	**0.12**
E07 [animal - sound]	0.15	0.31	0.25	**0.17**	0.03	0.00
E08 [animal - shelter]	0.25	0.29	0.39	0.00	**0.16**	0.09
E09 [things - color]	0.20	0.23	0.63	0.08	0.15	**0.21**
E10 [male - female]	0.62	0.28	0.33	0.66	**0.68**	**0.68**
Inflectional	0.63	0.22	0.23	**0.88**	0.87	**0.88**
I01 [noun - plural-reg]	0.69	0.13	0.16	0.84	0.84	**0.88**
I02 [noun - plural-irreg]	0.62	0.12	0.16	0.67	0.69	**0.75**
I03 [adj - comparative]	0.63	0.23	0.37	0.97	0.97	**1.00**
I04 [adj - superlative]	0.59	0.26	0.39	0.93	0.93	**0.97**
I05 [verb-inf - 3pSg]	0.65	0.26	0.33	**1.00**	**1.00**	**1.00**
I06 [verb-inf - Ving]	0.67	0.26	0.19	**0.84**	0.82	0.82
I07 [verb-inf - Ved]	0.66	0.25	0.20	**0.92**	0.90	0.88
I08 [verb-Ving - 3pSg]	0.56	0.17	0.31	0.90	**0.92**	0.90
I09 [verb-Ving - Ved]	0.64	0.18	0.19	**0.84**	**0.84**	0.82
I10 [verb-3pSg - Ved]	0.62	0.33	0.20	**0.90**	0.88	0.88
Lexicography	0.45	0.17	0.18	**0.19**	0.17	0.18
L01 [hypernyms - animals]	0.47	0.32	0.47	0.00	**0.05**	**0.05**
L02 [hypernyms - misc]	0.42	0.21	0.21	**0.29**	0.21	0.10
L03 [hyponyms - misc]	0.52	0.15	0.15	**0.19**	0.14	0.14
L04 [meronyms - substance]	0.35	0.16	0.24	**0.15**	0.09	0.11
L05 [meronyms - member]	0.36	0.15	0.15	**0.10**	0.08	0.08
L06 [meronyms - part]	0.34	0.14	0.12	**0.09**	**0.09**	0.02
L07 [synonyms - intensity]	0.51	0.17	0.16	0.24	0.26	**0.30**
L08 [synonyms - exact]	0.55	0.11	0.11	0.15	0.15	**0.22**
L09 [antonyms - gradable]	0.45	0.18	0.19	0.41	0.41	**0.43**
L10 [antonyms - binary]	0.50	0.14	0.15	0.21	0.17	**0.31**
Micro Mean	0.47	0.22	0.26	**0.49**	0.48	0.47
Macro Mean	0.46	0.22	0.26	**0.48**	0.47	0.45
Macro Std	0.14	0.07	0.12	0.31	0.31	**0.32**

Table 1: Detailed results for all combinations of FastText/Google News embeddings and Google Analogy and BATS analogies. In this table the cosine similarity is computed in the orthogonal complement. See the main paper for more details.

FastText

Google Analogy

	Mean Cosine Sim			Precision		
				IntCos		LRCos
	InterIntraL		IntraR	DensRay	SVM	
capital-common-countries	0.76	0.53	0.56	**1.00**	**1.00**	**1.00**
capital-world	0.75	0.44	0.51	0.98	0.98	**1.00**
city-in-state	0.71	0.51	0.63	**0.87**	0.85	0.85
currency	0.33	0.59	0.48	**0.54**	0.31	0.08
family	0.84	0.57	0.59	0.91	0.91	**0.95**
gram1-adjective-to-adverb	0.66	0.50	0.56	**0.88**	0.84	**0.88**
gram2-opposite	0.72	0.51	0.54	0.72	**0.83**	0.76
gram3-comparative	0.75	0.53	0.57	0.92	**0.95**	**0.95**
gram4-superlative	0.70	0.53	0.61	**1.00**	**1.00**	**1.00**
gram5-present-participle	0.76	0.43	0.48	**1.00**	**1.00**	**1.00**
gram6-nationality-adjective	0.70	0.55	0.54	**0.93**	**0.93**	**0.93**
gram7-past-tense	0.73	0.49	0.47	0.95	0.93	**0.97**
gram8-plural	0.80	0.41	0.42	**1.00**	**1.00**	**1.00**
gram9-plural-verbs	0.74	0.43	0.48	**1.00**	**1.00**	**1.00**
Micro Mean	0.73	0.48	0.53	**0.93**	0.92	**0.93**
Macro Mean	0.71	0.50	0.53	**0.91**	0.90	0.89
Macro Std	0.11	0.05	0.06	0.12	0.17	**0.23**

Google News

Google Analogy

	Mean Cosine Sim			Precision		
				IntCos		LRCos
	InterIntraL		IntraR	DensRay	SVM	
capital-common-countries	0.64	0.37	0.38	**0.91**	**0.91**	**0.91**
capital-world	0.64	0.34	0.36	**0.90**	**0.90**	**0.90**
city-in-state	0.59	0.37	0.49	0.85	**0.88**	0.87
currency	0.37	0.42	0.44	**0.67**	0.50	0.56
family	0.74	0.48	0.53	**0.87**	**0.87**	**0.87**
gram1-adjective-to-adverb	0.49	0.21	0.26	**0.78**	0.75	0.75
gram2-opposite	0.50	0.24	0.31	0.71	**0.75**	0.71
gram3-comparative	0.60	0.25	0.40	**0.97**	0.95	**0.97**
gram4-superlative	0.54	0.26	0.39	**0.94**	**0.94**	**0.94**
gram5-present-participle	0.70	0.20	0.21	**0.94**	0.91	**0.94**
gram6-nationality-adjective	0.72	0.41	0.41	**0.93**	**0.93**	**0.93**
gram7-past-tense	0.66	0.21	0.22	**0.88**	0.85	0.85
gram8-plural	0.73	0.21	0.23	**0.97**	0.89	**0.97**
gram9-plural-verbs	0.64	0.22	0.24	**0.93**	0.87	**0.93**
Micro Mean	0.62	0.31	0.36	**0.89**	0.87	0.88
Macro Mean	0.61	0.30	0.35	**0.88**	0.85	0.87
Macro Std	0.10	0.09	0.10	0.09	**0.11**	0.11

BATS

	Mean Cosine Sim			Precision		
				IntCos		LRCos
	InterIntraL		IntraR	DensRay	SVM	
Derivational	0.63	0.47	0.45	**0.81**	0.80	0.80
D01 [noun+less-reg]	0.51	0.36	0.45	**0.60**	**0.60**	**0.60**
D02 [un+adj-reg]	0.71	0.46	0.46	0.80	0.72	**0.84**
D03 [adj+ly-reg]	0.63	0.50	0.52	**0.90**	**0.90**	**0.90**
D04 [over+adj-reg]	0.63	0.45	0.45	0.66	**0.69**	0.62
D05 [adj+ness-reg]	0.63	0.49	0.51	**1.00**	0.96	0.92
D06 [re+verb-reg]	0.74	0.52	0.49	0.59	0.71	**0.76**
D07 [verb+able-reg]	0.57	0.48	0.45	**0.74**	0.63	0.63
D08 [verb+er-irreg]	0.55	0.48	0.41	**0.91**	0.86	0.79
D09 [verb+tion-irreg]	0.63	0.46	0.41	0.91	**0.93**	0.86
D10 [verb+ment-irreg]	0.64	0.46	0.42	**0.92**	0.90	0.90
Encyclopedia	0.48	0.43	0.55	0.41	0.43	**0.45**
E01 [country - capital]	0.63	0.47	0.41	0.96	0.96	**0.98**
E02 [country - language]	0.40	0.43	0.59	**0.33**	0.24	**0.33**
E03 [UK-city - county]	0.59	0.49	0.57	0.30	**0.42**	0.36
E04 [name - nationality]	0.28	0.39	0.64	0.49	0.51	**0.60**
E05 [name - occupation]	0.44	0.41	0.57	**0.75**	0.73	0.73
E06 [animal - young]	0.47	0.43	0.44	0.10	**0.15**	**0.15**
E07 [animal - sound]	0.37	0.43	0.40	**0.22**	0.17	**0.22**
E08 [animal - shelter]	0.44	0.42	0.51	0.02	**0.13**	**0.13**
E09 [things - color]	0.44	0.38	0.81	0.12	**0.22**	0.16
E10 [male - female]	0.73	0.43	0.43	0.76	0.71	**0.78**
Inflectional	0.75	0.48	0.51	**0.97**	**0.97**	**0.97**
I01 [noun - plural-reg]	0.79	0.39	0.41	**1.00**	**1.00**	**1.00**
I02 [noun - plural-irreg]	0.77	0.40	0.42	**0.84**	0.82	**0.84**
I03 [adj - comparative]	0.75	0.50	0.52	**1.00**	**1.00**	**1.00**
I04 [adj - superlative]	0.71	0.51	0.58	0.96	0.96	**1.00**
I05 [verb-inf - 3pSg]	0.77	0.52	0.53	**1.00**	**1.00**	**1.00**
I06 [verb-inf - Ving]	0.77	0.51	0.51	**0.96**	**0.96**	**0.96**
I07 [verb-inf - Ved]	0.75	0.51	0.55	**1.00**	**1.00**	**1.00**
I08 [verb-Ving - 3pSg]	0.70	0.48	0.51	**0.98**	**0.98**	**0.98**
I09 [verb-Ving - Ved]	0.73	0.50	0.54	**0.98**	0.96	**0.98**
I10 [verb-3pSg - Ved]	0.72	0.53	0.55	**1.00**	**1.00**	**1.00**
Lexicography	0.62	0.37	0.38	0.21	0.22	**0.26**
L01 [hypernyms - animals]	0.58	0.43	0.53	0.20	**0.37**	0.24
L02 [hypernyms - misc]	0.55	0.35	0.39	**0.23**	0.16	0.14
L03 [hyponyms - misc]	0.63	0.35	0.31	**0.33**	0.28	0.28
L04 [meronyms - substance]	0.53	0.36	0.44	0.10	0.15	**0.17**
L05 [meronyms - member]	0.58	0.36	0.36	**0.12**	0.10	**0.12**
L06 [meronyms - part]	0.53	0.31	0.30	0.04	**0.15**	0.09
L07 [synonyms - intensity]	0.67	0.37	0.37	0.27	0.32	**0.36**
L08 [synonyms - exact]	0.71	0.33	0.31	0.18	0.16	**0.22**
L09 [antonyms - gradable]	0.68	0.45	0.43	0.43	0.37	**0.55**
L10 [antonyms - binary]	0.72	0.40	0.40	0.18	0.18	**0.39**
Micro Mean	0.62	0.44	0.47	0.59	0.60	**0.61**
Macro Mean	0.62	0.44	0.47	0.60	0.60	**0.61**
Macro Std	0.12	0.06	0.09	**0.34**	0.33	0.32

BATS

	Mean Cosine Sim			Precision		
				IntCos		LRCos
	InterIntraL		IntraR	DensRay	SVM	
Derivational	0.44	0.21	0.20	**0.51**	0.48	0.44
D01 [noun+less-reg]	0.26	0.16	0.24	**0.10**	**0.10**	0.05
D02 [un+adj-reg]	0.47	0.17	0.20	**0.66**	0.58	0.58
D03 [adj+ly-reg]	0.48	0.17	0.22	0.76	**0.78**	0.76
D04 [over+adj-reg]	0.39	0.17	0.21	**0.41**	**0.41**	0.30
D05 [adj+ness-reg]	0.47	0.21	0.26	**0.75**	0.70	0.65
D06 [re+verb-reg]	0.56	0.29	0.28	**0.69**	0.61	0.58
D07 [verb+able-reg]	0.38	0.25	0.20	**0.28**	0.22	0.19
D08 [verb+er-irreg]	0.30	0.24	0.17	**0.12**	0.10	0.07
D09 [verb+tion-irreg]	0.51	0.22	0.15	**0.61**	**0.61**	0.51
D10 [verb+ment-irreg]	0.47	0.24	0.15	**0.54**	0.50	0.44
Encyclopedia	0.35	0.29	0.42	**0.35**	0.32	0.34
E01 [country - capital]	0.61	0.35	0.32	**0.90**	**0.90**	**0.90**
E02 [country - language]	0.36	0.31	0.45	**0.43**	0.21	0.36
E03 [UK-city - county]	0.41	0.36	0.52	**0.14**	0.12	**0.14**
E04 [name - nationality]	0.20	0.20	0.39	**0.26**	0.15	**0.26**
E05 [name - occupation]	0.33	0.21	0.40	**0.60**	0.57	0.52
E06 [animal - young]	0.34	0.36	0.38	0.09	**0.12**	**0.12**
E07 [animal - sound]	0.15	0.31	0.25	**0.08**	0.00	0.00
E08 [animal - shelter]	0.25	0.29	0.39	0.07	**0.14**	0.09
E09 [things - color]	0.20	0.23	0.63	0.19	0.19	**0.21**
E10 [male - female]	0.62	0.28	0.33	**0.68**	**0.68**	**0.68**
Inflectional	0.63	0.22	0.23	**0.88**	**0.88**	**0.88**
I01 [noun - plural-reg]	0.69	0.13	0.16	0.86	0.84	**0.88**
I02 [noun - plural-irreg]	0.62	0.12	0.16	0.69	0.71	**0.75**
I03 [adj - comparative]	0.63	0.23	0.37	0.97	**1.00**	**1.00**
I04 [adj - superlative]	0.59	0.26	0.39	**0.97**	**0.97**	**0.97**
I05 [verb-inf - 3pSg]	0.65	0.26	0.33	**1.00**	**1.00**	**1.00**
I06 [verb-inf - Ving]	0.67	0.26	0.19	**0.86**	0.82	0.82
I07 [verb-inf - Ved]	0.66	0.25	0.20	**0.92**	0.90	0.88
I08 [verb-Ving - 3pSg]	0.56	0.17	0.31	**0.90**	**0.90**	**0.90**
I09 [verb-Ving - Ved]	0.64	0.18	0.19	**0.84**	0.82	0.82
I10 [verb-3pSg - Ved]	0.62	0.33	0.20	**0.88**	**0.88**	**0.88**
Lexicography	0.45	0.17	0.18	**0.19**	0.17	0.18
L01 [hypernyms - animals]	0.47	0.32	0.47	**0.08**	**0.08**	0.05
L02 [hypernyms - misc]	0.42	0.21	0.21	**0.26**	**0.26**	0.10
L03 [hyponyms - misc]	0.52	0.15	0.15	**0.19**	0.14	0.14
L04 [meronyms - substance]	0.35	0.16	0.24	0.09	0.09	**0.11**
L05 [meronyms - member]	0.36	0.15	0.15	**0.10**	0.06	0.08
L06 [meronyms - part]	0.34	0.14	0.12	**0.09**	**0.09**	0.02
L07 [synonyms - intensity]	0.51	0.17	0.16	0.26	0.26	**0.30**
L08 [synonyms - exact]	0.55	0.11	0.11	0.15	0.15	**0.22**
L09 [antonyms - gradable]	0.45	0.18	0.19	**0.45**	0.41	0.43
L10 [antonyms - binary]	0.50	0.14	0.15	0.19	0.17	**0.31**
Micro Mean	0.47	0.22	0.26	**0.49**	0.47	0.47
Macro Mean	0.46	0.22	0.26	**0.48**	0.46	0.45
Macro Std	0.14	0.07	0.12	0.32	**0.32**	0.32

Table 2: Detailed results for all combinations of FastText/Google News embeddings and Google Analogy and BATS analogies. In this table the cosine similarity is computed in the original space. See the main paper for more details.

Chapter 5

SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings

SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings

Masoud Jalili Sabet[*1], **Philipp Dufter**[*1], **François Yvon**[2], **Hinrich Schütze**[1]
[1] Center for Information and Language Processing (CIS), LMU Munich, Germany
[2] Université Paris-Saclay, CNRS, LIMSI, France
{masoud,philipp}@cis.lmu.de, francois.yvon@limsi.fr

Abstract

Word alignments are useful for tasks like statistical and neural machine translation (NMT) and cross-lingual annotation projection. Statistical word aligners perform well, as do methods that extract alignments jointly with translations in NMT. However, most approaches require parallel training data, and quality decreases as less training data is available. We propose word alignment methods that require no parallel data. The key idea is to leverage multilingual word embeddings – both static and contextualized – for word alignment. Our multilingual embeddings are created from monolingual data only without relying on any parallel data or dictionaries. We find that alignments created from embeddings are superior for four and comparable for two language pairs compared to those produced by traditional statistical aligners – even with abundant parallel data; e.g., contextualized embeddings achieve a word alignment F_1 for English-German that is 5 percentage points higher than eflomal, a high-quality statistical aligner, trained on 100k parallel sentences.

1 Introduction

Word alignments are essential for statistical machine translation and useful in NMT, e.g., for imposing priors on attention matrices (Liu et al., 2016; Chen et al., 2016; Alkhouli and Ney, 2017; Alkhouli et al., 2018) or for decoding (Alkhouli et al., 2016; Press and Smith, 2018). Further, word alignments have been successfully used in a range of tasks such as typological analysis (Lewis and Xia, 2008; Östling, 2015b), annotation projection (Yarowsky et al., 2001; Padó and Lapata, 2009; Asgari and Schütze, 2017; Huck et al., 2019) and creating multilingual embeddings (Guo et al., 2016; Ammar et al., 2016; Dufter et al., 2018).

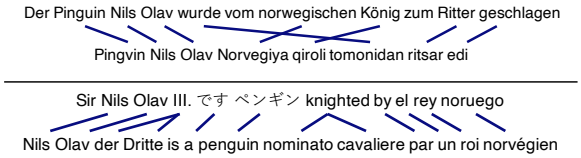


Figure 1: Our method does not rely on parallel training data and can align distant language pairs (German-Uzbek, top) and even mixed sentences (bottom). Example sentence is manually created. Algorithm: Itermax.

Statistical word aligners such as the IBM models (Brown et al., 1993) and their implementations Giza++ (Och and Ney, 2003), fast-align (Dyer et al., 2013), as well as newer models such as eflomal (Östling and Tiedemann, 2016) are widely used for alignment. With the rise of NMT (Bahdanau et al., 2014), attempts have been made to interpret attention matrices as soft word alignments (Cohn et al., 2016; Koehn and Knowles, 2017; Ghader and Monz, 2017). Several methods create alignments from attention matrices (Peter et al., 2017; Zenkel et al., 2019) or pursue a multitask approach for alignment and translation (Garg et al., 2019). However, most systems require parallel data (in sufficient amount to train high quality NMT systems) and their performance deteriorates when parallel text is scarce (Tables 1–2 in (Och and Ney, 2003)).

Recent unsupervised multilingual embedding algorithms that use only non-parallel data provide high quality static (Artetxe et al., 2018; Conneau et al., 2018) and contextualized embeddings (Devlin et al., 2019; Conneau et al., 2020). *Our key idea is to leverage these embeddings for word alignments – by extracting alignments from similarity matrices induced from embeddings – without relying on parallel data.* Requiring no or little parallel data is advantageous, e.g., in the low-resource case and in domain-specific settings without parallel data. A lack of parallel data cannot be easily

[*] Equal contribution - random order.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 1627–1643
November 16 - 20, 2020.

remedied: mining parallel sentences is possible (Schwenk et al., 2019) but assumes that comparable, monolingual corpora contain parallel sentences. Further, we find that large amounts of mined parallel data do not necessarily improve alignment quality.

Our main **contribution** is that we show that *word alignments obtained from multilingual pretrained language models are superior for four and comparable for two language pairs, compared to strong statistical word aligners like eflomal even in high resource scenarios.* Additionally, **(1)** we introduce three new alignment methods based on the matrix of embedding similarities and two extensions that handle null words and integrate positional information. They permit a flexible tradeoff of recall and precision. **(2)** We provide evidence that subword processing is beneficial for aligning rare words. **(3)** We bundle the source code of our methods in a tool called *SimAlign*, which is available.[1] An interactive online demo is available.[2]

2 Methods

2.1 Alignments from Similarity Matrices

We propose three methods to obtain alignments from similarity matrices. Argmax is a simple baseline, IterMax a novel iterative algorithm, and Match a graph-theoretical method based on identifying matchings in a bipartite graph.

Consider parallel sentences $s^{(e)}, s^{(f)}$, with lengths l_e, l_f in languages e, f. Assume we have access to some embedding function $\mathcal{E}$ that maps each word in a sentence to a d-dimensional vector, i.e., $\mathcal{E}(s^{(k)}) \in \mathbb{R}^{l_k \times d}$ for $k \in \{e, f\}$. Let $\mathcal{E}(s^{(k)})_i$ denote the vector of the i-th word in sentence $s^{(k)}$. For static embeddings $\mathcal{E}(s^{(k)})_i$ depends only on the word i in language k whereas for contextualized embeddings the vector depends on the full context $s^{(k)}$. We define the *similarity matrix* as the matrix $S \in [0,1]^{l_e \times l_f}$ induced by the embeddings where $S_{ij} := \text{sim}\left(\mathcal{E}(s^{(e)})_i, \mathcal{E}(s^{(f)})_j\right)$ is some normalized measure of similarity, e.g., cosine-similarity normalized to be between 0 and 1. We now describe our methods for extracting alignments from S, i.e., obtaining a binary matrix $A \in \{0,1\}^{l_e \times l_f}$.

Argmax. A simple baseline is to align i and j when $s_i^{(e)}$ is the most similar word to $s_j^{(f)}$ and vice-versa. That is, we set $A_{ij} = 1$ if

$$(i = \arg\max_l S_{l,j}) \wedge (j = \arg\max_l S_{i,l})$$

and $A_{ij} = 0$ otherwise. In case of ties, which are unlikely in similarity matrices, we choose the smaller index. If all entries in a row i or column j of S are 0 we set $A_{ij} = 0$ (this case can appear in Itermax). Similar methods have been applied to co-occurrences (Melamed, 2000) ("competitive linking"), Dice coefficients (Och and Ney, 2003) and attention matrices (Garg et al., 2019).

Algorithm 1 Itermax.

procedure ITERMAX(S, $n_{\max}$, $\alpha \in [0,1]$)
  $A, M = \text{zeros\_like}(S)$
  for $n \in [1, \ldots, n_{\max}]$ **do**
    $\forall i, j$:

$$M_{ij} = \begin{cases} 1 \text{ if } \max\left(\sum_{l=0}^{l_e} A_{lj}, \sum_{l=0}^{l_f} A_{il}\right) = 0 \\ 0 \text{ if } \min\left(\sum_{l=0}^{l_e} A_{lj}, \sum_{l=0}^{l_f} A_{il}\right) > 0 \\ \alpha \text{ otherwise} \end{cases}$$

    $A_{\text{to\_add}} = \text{get\_argmax\_alignments}(S \odot M)$
    $A = A + A_{\text{to\_add}}$
  end for
  return A
end procedure

Figure 2: Description of the Itermax algorithm. *zeros_like* yields a matrix with zeros and with same shape as the input, *get_argmax_alignments* returns alignments obtained using the Argmax Method, $\odot$ is elementwise multiplication.

Itermax. There are many sentences for which Argmax only identifies few alignment edges because mutual argmaxes can be rare. As a remedy, we apply Argmax iteratively. Specifically, we modify the similarity matrix conditioned on the alignment edges found in a previous iteration: if two words i and j have *both* been aligned, we zero out the similarity. Similarly, if *neither* is aligned we leave the similarity unchanged. In case only one of them is aligned, we multiply the similarity with a discount factor $\alpha \in [0,1]$. Intuitively, this encourages the model to focus on unaligned word pairs. However, if the similarity with an already aligned word is exceptionally high, the model can add an additional edge. Note that this explicitly allows one token to be aligned to multiple other tokens. For details on the algorithm see Figure 2.

Match. Argmax finds a local, not a global optimum and Itermax is a greedy algorithm. To find global optima, we frame alignment as an assign-

[1] https://github.com/cisnlp/simalign
[2] https://simalign.cis.lmu.de/

ment problem: we search for a maximum-weight maximal matching (e.g., (Kuhn, 1955)) in the bipartite weighted graph which is induced by the similarity matrix. This optimization problem is defined by

$$A^* = \text{argmax}_{A \in \{0,1\}^{l_e \times l_f}} \sum_{i=1}^{l_e} \sum_{j=1}^{l_f} A_{ij} S_{ij}$$

subject to A being a matching (i.e., each node has at most one edge) that is maximal (i.e., no additional edge can be added). There are known algorithms to solve the above problem in polynomial time (e.g., (Galil, 1986)).

Note that alignments generated with the match method are inherently bidirectional. None of our methods require additional symmetrization as post-processing.

2.2 Distortion and Null Extensions

Distortion Correction [Dist]. Distortion, as introduced in IBM Model 2, is essential for alignments based on non-contextualized embeddings since the similarity of two words is solely based on their surface form, independent of position. To penalize high distortions, we multiply the similarity matrix S componentwise with

$$P_{i,j} = 1 - \kappa \left(i/l_e - j/l_f\right)^2,$$

where κ is a hyperparameter to scale the distortion matrix P between $[(1-\kappa), 1]$. We use $\kappa = 0.5$. See supplementary for different values. We can interpret this as imposing a locality-preserving prior: given a choice, a word should be aligned to a word with a similar relative position ($\left(i/l_e - j/l_f\right)^2$ close to 0) rather than a more distant word (large $\left(i/l_e - j/l_f\right)^2$).

Null. Null words model untranslated words and are an important part of alignment models. We propose to model null words as follows: if a word is not particularly similar to any of the words in the target sentence, we do not align it. Specifically, given an alignment matrix A, we remove alignment edges when the normalized entropy of the similarity distribution is above a threshold τ, a hyperparameter. We use normalized entropy (i.e., entropy divided by the log of sentence length) to account for different sentence lengths; i.e., we set $A_{ij} = 0$ if

$$\min\left(-\frac{\sum_{k=1}^{l_f} S^h_{ik} \log S^h_{ik}}{\log l_f}, -\frac{\sum_{k=1}^{l_e} S^v_{kj} \log S^v_{kj}}{\log l_e}\right) > \tau,$$

where $S^h_{ik} := S_{ik} / \sum_{m=1}^{l_f} S_{im}$, and $S^v_{kj} := S_{kj} / \sum_{m=1}^{l_e} S_{mj}$. As the ideal value of τ depends on the actual similarity scores we set τ to a percentile of the entropy values of the similarity distribution across all aligned edges (we use the 95th percentile). Different percentiles are in the supplementary.

3 Experiments

3.1 Embedding Learning

Static. We train monolingual embeddings with fastText (Bojanowski et al., 2017) for each language on its Wikipedia. We then use VecMap (Artetxe et al., 2018) to map the embeddings into a common multilingual space. Note that this algorithm works without any crosslingual supervision (e.g., multilingual dictionaries). We use the same procedure for word and subword levels. We use the label **fastText** to refer to these embeddings as well as the alignments induced by them.

Contextualized. We use the multilingual BERT model (mBERT).[3] It is pretrained on the 104 largest Wikipedia languages. This model only provides embeddings at the subword level. To obtain a word embedding, we simply average the vectors of its subwords. We consider word representations from all 12 layers as well as the concatenation of all layers. Note that the model is not finetuned. We denote this method as mBERT[i] (when using embeddings from the i-th layer, where 0 means using the non-contextualized initial embedding layer) and mBERT[conc] (for concatenation).

In addition, we use XLM-RoBERTa base (Conneau et al., 2020), which is pretrained on 100 languages on cleaned CommonCrawl data (Wenzek et al., 2020). We denote alignments obtained using the embeddings from the i-th layer by XLM-R[i].

3.2 Word and Subword Alignments

We investigate both alignments between subwords such as wordpiece (Schuster and Nakajima, 2012) (which are widely used for contextualized language models) and words. We refer to computing alignment edges between words as *word level* and between subwords as *subword level*. Note that gold standards are all word-level. In order to evaluate alignments obtained at the subword level we convert subword to word alignments using the heuristic "two words are aligned if any of their subwords are

[3] https://github.com/google-research/bert/blob/master/multilingual.md

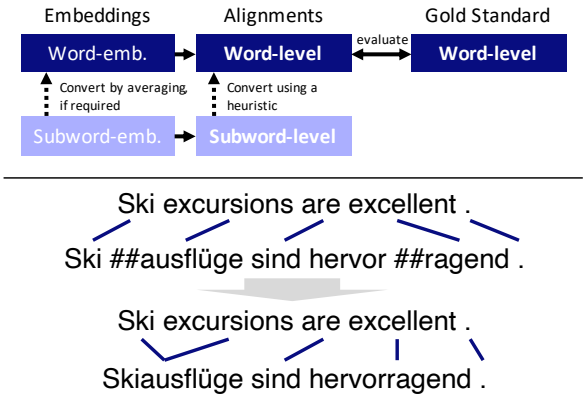


Figure 3: Subword alignments are always converted to word alignments for evaluation.

aligned" (see Figure 3). As a result a single word can be aligned with multiple other words.

For the *word* level, we use the NLTK tokenizer (Bird et al., 2009) (e.g., for tokenizing Wikipedia in order to train fastText). For the *subword* level, we generally use multilingual BERT's vocabulary[3] and BERT's wordpiece tokenizer. For XLM-R we use the XLM-R subword vocabulary. Since gold standards are already tokenized, they do not require additional tokenization.

3.3 Baselines

We compare to three popular statistical alignment models that all require parallel training data. **fast-align/IBM2** (Dyer et al., 2013) is an implementation of an alignment algorithm based on IBM Model 2. It is popular because of its speed and high quality. **eflomal**[4] (based on efmaral by Östling and Tiedemann (2016)), a Bayesian model with Markov Chain Monte Carlo inference, is claimed to outperform fast-align on speed and quality. Further we use the widely used software package **Giza++/IBM4** (Och and Ney, 2003), which implements IBM alignment models. We use its standard settings: 5 iterations each for the HMM model, IBM Models 1, 3 and 4 with $p_0 = 0.98$.

Symmetrization. Probabilistic word alignment models create forward and backward alignments and then symmetrize them (Och and Ney, 2003; Koehn et al., 2005). We compared the symmetrization methods grow-diag-final-and (GDFA) and intersection and found them to perform comparably; see supplementary. We use GDFA throughout the paper.

[4] github.com/robertostling/eflomal

3.4 Evaluation Measures

Given a set of predicted alignment edges A and a set of sure, possible gold standard edges S, P (where $S \subset P$), we use the following evaluation measures:

$$\text{prec} = \frac{|A \cap P|}{|A|}, \text{rec} = \frac{|A \cap S|}{|S|},$$
$$F_1 = \frac{2 \text{ prec rec}}{\text{prec} + \text{rec}},$$
$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|},$$

where $|\cdot|$ denotes the cardinality of a set. This is the standard evaluation (Och and Ney, 2003).

3.5 Data

Our **test data** are a diverse set of 6 language pairs: Czech, German, Persian, French, Hindi and Romanian, always paired with English. See Table 11 for corpora and supplementary for URLs.

For our baselines requiring parallel training data (i.e., eflomal, fast-align and Giza++) we select additional parallel **training data** that is consistent with the target domain where available. See Table 11 for the corpora. Unless indicated otherwise we use the whole parallel training data. Figure 5 shows the effect of using more or less training data.

Given the large amount of possible experiments when considering 6 language pairs we do not have space to present all numbers for all languages. If we show results for only one pair, we choose ENG-DEU as it is an established and well-known dataset (EuroParl). If we show results for more languages we fall back to DEU, CES and HIN, to show effects on a mid-resource morphologically rich language (CES) and a low-resource language written in a different script (HIN).

4 Results

4.1 Embedding Layer

Figure 4 shows a parabolic trend across layers of mBERT and XLM-R. We use layer 8 in this paper because it has best performance. This is consistent with other work (Hewitt and Manning, 2019; Tenney et al., 2019): in the first layers the contextualization is too weak for high-quality alignments while the last layers are too specialized on the pre-training task (masked language modeling).

Lang.	Gold Standard	Gold St. Size	$\|S\|$	$\|P \setminus S\|$	Parallel Data	Parallel Data Size	Wikipedia Size
ENG-CES	(Mareček, 2008)	2500	44292	23132	EuroParl (Koehn, 2005)	646k	8M
ENG-DEU	EuroParl-based[a]	508	9612	921	EuroParl (Koehn, 2005)	1920k	48M
ENG-FAS	(Tavakoli and Faili, 2014)	400	11606	0	TEP (Pilevar et al., 2011)	600k	5M
ENG-FRA	WPT2003, (Och and Ney, 2000),	447	4038	13400	Hansards (Germann, 2001)	1130k	32M
ENG-HIN	WPT2005[b]	90	1409	0	Emille (McEnery et al., 2000)	3k	1M
ENG-RON	WPT2005[b]	203	5033	0	Constitution, Newspaper[b]	50k	3M

[a] www-i6.informatik.rwth-aachen.de/goldAlignment/
[b] http://web.eecs.umich.edu/~mihalcea/wpt05/

Table 1: Overview of datasets. "Lang." uses ISO 639-3 language codes. "Size" refers to the number of sentences. "Parallel Data Size" refers to the number of parallel sentences in addition to the gold alignments that is used for training the baselines. Our sentence tokenized version of the English Wikipedia has 105M sentences.

	Method	ENG-CES F_1	ENG-CES AER	ENG-DEU F_1	ENG-DEU AER	ENG-FAS F_1	ENG-FAS AER	ENG-FRA F_1	ENG-FRA AER	ENG-HIN F_1	ENG-HIN AER	ENG-RON F_1	ENG-RON AER
Prior Work	(Östling, 2015a) Bayesian							**.94**	**.06**	.57	.43	**.73**	**.27**
	(Östling, 2015a) Giza++							.92	.07	.51	.49	.72	.28
	(Legrand et al., 2016) Ensemble Method	.81	.16					.71	.10				
	(Östling and Tiedemann, 2016) efmaral							.93	.08	.53	.47	.72	.28
	(Östling and Tiedemann, 2016) fast-align							.86	.15	.33	.67	.68	.33
	(Zenkel et al., 2019) Giza++				.21				**.06**				.28
	(Garg et al., 2019) Multitask				.20				.08				
Baselines Word	fast-align/IBM2	.76	.25	.71	.29	.57	.43	.86	.15	.34	.66	.68	.33
	Giza++/IBM4	.75	.26	.77	.23	.51	.49	.92	.09	.45	.55	.69	.31
	eflomal	.85	.15	.77	.23	.61	.39	.93	.08	.51	.49	.71	.29
Baselines Subword	fast-align/IBM2	.78	.23	.71	.30	.58	.42	.85	.16	.38	.62	.68	.32
	Giza++/IBM4	.82	.18	.78	.22	.57	.43	.92	.09	.48	.52	.69	.32
	eflomal	.84	.17	.76	.24	.63	.37	.91	.09	.52	.48	.72	.28
This Work Word	fastText - Argmax	.70	.30	.60	.40	.50	.50	.77	.22	.49	.52	.47	.53
	mBERT[8] - Argmax	**.87**	**.13**	.79	.21	.67	.33	**.94**	**.06**	.54	.47	.64	.36
	XLM-R[8] - Argmax	**.87**	**.13**	.79	.21	.70	.30	.93	**.06**	.59	.41	.70	.30
This Work Subword	fastText - Argmax	.58	.42	.56	.44	.09	.91	.73	.26	.04	.96	.43	.58
	mBERT[8] - Argmax	.86	.14	**.81**	**.19**	.67	.33	**.94**	**.06**	.55	.45	.65	.35
	XLM-R[8] - Argmax	**.87**	**.13**	**.81**	**.19**	**.71**	**.29**	.93	.07	**.61**	**.39**	.71	.29

Table 2: Comparison of our methods, baselines and prior work in unsupervised word alignment. Best result per column in bold. A detailed version of the table with precision/recall and Itermax/Match results is in supplementary.

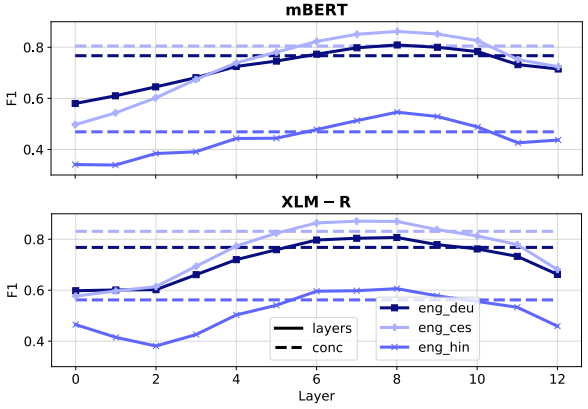


Figure 4: Word alignment performance across layers of mBERT (top) and XLM-R (bottom). Results are F_1 with Argmax at the subword level.

4.2 Comparison with Prior Work

Contextual Embeddings. Table 2 shows that mBERT and XLM-R consistently perform well with the Argmax method. XLM-R yields mostly higher values than mBERT. Our three baselines, eflomal, fast-align and Giza++, are always outperformed (except for RON). We outperform all prior work except for FRA where we match the performance and RON. This comparison is not entirely fair because methods relying on parallel data have access to the parallel sentences of the test data during training whereas our methods do not.

Romanian might be a special case as it exhibits a large amount of many to one links and further lacks determiners. How determiners are handled in the gold standard depends heavily on the annotation guidelines. Note that one of our settings, XLM-R[8] with Itermax at the subword level, has an F1 of .72 for ENG-RON, which comes very close to the performance by (Östling, 2015a) (see Table 3).

In summary, extracting alignments from similarity matrices is a very simple and efficient method that performs surprisingly strongly. It outperforms strong statistical baselines and most prior work in unsupervised word alignment for CES, DEU, FAS and HIN and is comparable for FRA and RON. We attribute this to the strong contextualization in mBERT and XLM-R.

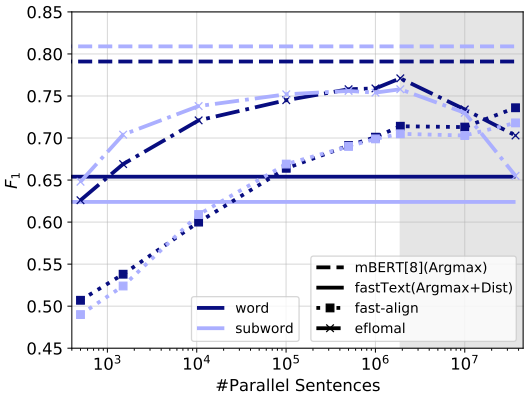


Figure 5: Learning curves of fast-align/eflomal vs. embedding-based alignments. Results shown are F_1 for ENG-DEU, contrasting subword and word representations. Up to 1.9M parallel sentences we use EuroParl. To demonstrate the effect with abundant parallel data we add up to 37M *additional* parallel sentences from ParaCrawl (Esplà et al., 2019) (see grey area).

Static Embeddings. fastText shows a solid performance on word level, which is worse but comes close to fast-align and outperforms it for HIN. We consider this surprising as fastText did not have access to parallel data or any multilingual signal. VecMap can also be used with crosslingual dictionaries. We expect this to boost performance and fastText could then become a viable alternative to fast-align.

Amount of Parallel Data. Figure 5 shows that fast-align and eflomal get better with more training data with eflomal outperforming fast-align, as expected. However, even with 1.9M parallel sentences mBERT outperforms both baselines. When adding up to 37M additional parallel sentences from ParaCrawl (Esplà et al., 2019) performance for fast-align increases slightly, however, eflomal decreases (grey area in plot). ParaCrawl contains mined parallel sentences whose lower quality probably harms eflomal. fastText (with distortion) is competitive with eflomal for fewer than 1000 parallel sentences and outperforms fast-align even with 10k sentences. Thus for very small parallel corpora (<10k sentences) using fastText embeddings is an alternative to fast-align.

The main takeaway from Figure 5 is that mBERT-based alignments, a method that does not need any parallel training data, outperforms state-of-the-art aligners like eflomal for ENG-DEU, even in the very high resource case.

Emb.	Method	ENG-CES	ENG-DEU	ENG-FAS	ENG-FRA	ENG-HIN	ENG-RON
mBERT[8]	Argmax	**.86**	**.81**	.67	**.94**	.55	.65
	Itermax	**.86**	**.81**	**.70**	.93	**.58**	**.69**
	Match	.82	.78	.67	.90	**.58**	.67
XLM-R[8]	Argmax	**.87**	**.81**	.71	**.93**	.61	.71
	Itermax	.86	.80	**.72**	.92	**.62**	**.72**
	Match	.81	.76	.68	.88	.60	.70

Table 3: Comparison of our three proposed methods across all languages for the best embeddings from Table 2: mBERT[8] and XLM-R[8]. We show F_1 at the subword level. Best result per embedding type in bold.

Emb.	$n_{\max}$	α	ENG-DEU Prec.	Rec.	F_1	AER	ENG-CES Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER
mBERT[8]	1	-	**.92**	.69	.79	.21	**.95**	.80	**.87**	**.13**	**.84**	.39	.54	.47
	2	.90	.85	.77	**.81**	**.19**	.87	.87	**.87**	.14	.75	.47	.58	.42
		.95	.83	.80	**.81**	**.19**	.85	.89	**.87**	**.13**	.73	.48	.58	.42
		1	.77	.79	.78	.22	.80	.86	.83	.17	.63	.46	.53	.47
	3	.90	.81	.80	.80	.20	.83	.88	.85	.15	.70	.49	.57	.43
		.95	.78	**.83**	**.81**	.20	.81	**.91**	.86	.15	.68	**.52**	**.59**	**.41**
		1	.73	**.83**	.77	.23	.76	**.91**	.82	.18	.58	.51	.54	.46
fastText	1	-	**.81**	.48	.60	.40	**.86**	.59	.70	.30	**.75**	.36	.49	.52
	2	.90	.69	.56	**.62**	**.38**	.74	.69	**.72**	**.29**	.63	.42	**.51**	**.49**
		.95	.66	.56	.61	.39	.71	.69	.70	.30	.59	.41	.48	.52
		1	.59	.55	.57	.43	.62	.65	.63	.37	.53	.39	.45	.55
	3	.90	.63	**.59**	.61	.39	.67	.72	.70	.31	.57	.43	.49	.51
		.95	.59	**.59**	.59	.41	.63	**.73**	.68	.33	.53	**.44**	.48	.52
		1	.53	.58	.55	.45	.55	.70	.62	.39	.48	.43	.45	.55

Table 4: Itermax with different number of iterations ($n_{\max}$) and different α. Results are at the word level.

4.3 Additional Methods and Extensions

We already showed that Argmax yields alignments that are competitive with the state of the art. In this section we compare all our proposed methods and extensions more closely.

Itermax. Table 4 shows results for Argmax (i.e., 1 Iteration) as well as Itermax (i.e., 2 or more iterations of Argmax). As expected, with more iterations precision drops in favor of recall. Overall, Itermax achieves higher F_1 scores for the three language pairs (equal for ENG-CES) both for mBERT[8] and fastText embeddings. For Hindi the performance increase is the highest. We hypothesize that for more distant languages Itermax is more beneficial as similarity between wordpieces may be generally lower, thus exhibiting fewer mutual argmaxes. For the rest of the paper if we use Itermax we use 2 Iterations with $\alpha = 0.9$ as it exhibits best performance (5 out of 6 wins in Table 4).

Argmax/Itermax/Match. In Table 3 we compare our three proposed methods in terms of F_1 across all languages. We chose to show the two

Emb.	Method	ENG-DEU Prec.	Rec.	F_1	AER	ENG-CES Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER
fastText	Argmax	.81	.48	.60	.40	.86	.59	.70	.30	**.75**	**.36**	**.49**	**.52**
	+Dist	**.84**	**.54**	**.65**	**.35**	**.89**	**.68**	**.77**	**.23**	.64	.30	.41	.59
	+Null	.81	.46	.59	.41	.86	.56	.68	.32	.74	.34	.46	.54
	Itermax	.69	.56	.62	.38	.74	.69	.72	.29	**.63**	**.42**	**.51**	**.49**
	+Dist	**.71**	**.62**	**.66**	**.34**	**.75**	**.76**	**.76**	**.25**	.54	.37	.44	.57
	+Null	.69	.53	.60	.40	.74	.66	.70	.30	**.63**	.40	.49	.51
	Match	.60	.58	.59	.41	.65	.71	.68	.32	.55	**.43**	**.48**	**.52**
	+Dist	**.67**	**.64**	**.65**	**.35**	**.72**	**.78**	**.75**	**.25**	.50	.39	.43	.57
	+Null	.61	.56	.58	.42	.66	.69	.67	.33	**.56**	.41	**.48**	**.52**
mBERT[8]	Argmax	.92	**.69**	**.79**	**.21**	**.95**	**.80**	**.87**	**.13**	.84	**.39**	**.54**	**.47**
	+Dist	.91	.67	.77	.23	.93	.79	.85	.15	.68	.29	.41	.59
	+Null	**.93**	.67	.78	.22	**.95**	.77	.85	.15	**.85**	.38	.53	**.47**
	Itermax	.85	**.77**	**.81**	**.19**	.87	**.87**	**.87**	**.14**	.75	**.47**	**.58**	**.43**
	+Dist	.82	.75	.79	.21	.84	.85	.85	.15	.56	.34	.43	.58
	+Null	**.86**	.75	.80	.20	**.88**	.84	.86	**.14**	**.76**	.45	.57	**.43**
	Match	.78	**.74**	**.76**	**.24**	.81	**.85**	**.83**	**.17**	.67	**.52**	**.59**	**.42**
	+Dist	.75	.71	.73	.27	.79	.83	.81	.20	.45	.35	.39	.61
	+Null	**.80**	.73	**.76**	**.24**	**.83**	.83	**.83**	**.17**	**.68**	.51	.58	**.42**

Table 5: Analysis of Null and Distortion Extensions. All alignments are obtained at word-level. Best result per embedding type and method in bold.

best performing settings from Table 2: mBERT[8] and XLM-R[8] at the subword level. Itermax performs slightly better than Argmax with 6 wins, 4 losses and 2 ties. Itermax seems to help more for more distant languages such as FAS, HIN and RON, but harms for FRA. Match has the lowest F_1, but generally exhibits a higher recall (see e.g., Table 5).

Null and Distortion Extensions. Table 5 shows that Argmax and Itermax generally have higher precision, whereas Match has higher recall. Adding Null almost always increases precision, but at the cost of recall, resulting mostly in a lower F_1 score. Adding a distortion prior boosts performance for static embeddings, e.g., from .70 to .77 for ENG-CES Argmax F_1 and similarly for ENG-DEU. For Hindi a distortion prior is harmful. Dist has little and sometimes harmful effects on mBERT indicating that mBERT's contextualized representations already match well across languages.

Summary. Argmax and Itermax exhibit the best and most stable performance. For most language pairs Itermax is recommended. If high recall alignments are required, Match is the recommended algorithm. Except for HIN, a distortion prior is beneficial for static embeddings. Null should be applied when one wants to push precision even higher (e.g., for annotation projection).

4.4 Words and Subwords

Table 2 shows that subword processing slightly outperforms word-level processing for most methods. Only fastText is harmed by subword processing.

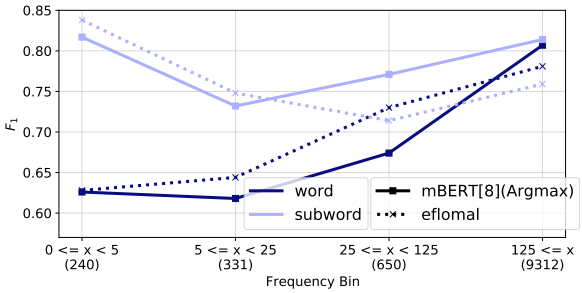


Figure 6: Results for different frequency bins on ENG-DEU. An edge in S, P, or A is attributed to exactly one bin based on the minimum frequency of the involved words (denoted by x). Number of gold edges in brackets. Eflomal is trained on all 1.9M parallel sentences. Frequencies are computed on the same corpus.

		ADJ	ADP	ADV	AUX	NOUN	PRON	VERB
eflomal	Word	**0.83**	0.69	**0.72**	0.63	0.85	0.79	0.63
	Subword	0.82	0.68	0.71	0.57	0.85	0.77	0.62
mBERT[8]	Word	0.79	0.74	0.71	0.71	0.81	**0.84**	**0.69**
	Subword	0.81	**0.75**	**0.72**	**0.72**	**0.87**	**0.84**	**0.69**

Table 6: Alignment performance (F_1) on ENG-DEU for POS. We use mBERT[8](Argmax) and Eflomal trained on 1.9M parallel sentences on the word level.

We use VecMap to match (sub)word distributions across languages. We hypothesize that it is harder to match subword than word distributions – this effect is strongest for Persian and Hindi, probably due to different scripts and thus different subword distributions. Initial experiments showed that adding supervision in form of a dictionary helps restore performance. We will investigate this in future work.

We hypothesize that subword processing is beneficial for aligning rare words. To show this, we compute our evaluation measures for different frequency bins. More specifically, we only consider gold standard alignment edges for the computation where at least one of the member words has a certain frequency in a reference corpus (in our case all 1.9M lines from the ENG-DEU EuroParl corpus). That is, we only consider the edge (i, j) in A, S or P if the minimum of the source and target word frequency is in $[\gamma_l, \gamma_u)$ where γ_l and γ_u are bin boundaries.

Figure 6 shows F_1 for different frequency bins. For rare words both eflomal and mBERT show a severely decreased performance at the word level, but not at the subword level. Thus, subword processing is indeed beneficial for rare words.

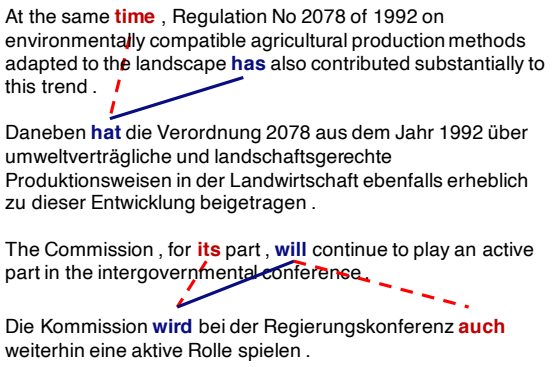


Figure 7: Example alignment of auxiliary verbs. Same setting as in Table 6. Solid lines: mBERT's alignment, identical to the gold standard. Dashed lines: eflomal's incorrect alignment.

4.5 Part-Of-Speech Analysis

To analyze the performance with respect to different part-of-speech (POS) tags, the ENG-DEU gold standard was tagged with the Stanza toolkit (Qi et al., 2020). We evaluate the alignment performance for each POS tag by only considering the alignment edges where at least one of their member words has this tag. Table 6 shows results for frequent POS tags. Compared to eflomal, mBERT aligns auxiliaries, pronouns and verbs better. The relative position of auxiliaries and verbs in German can diverge strongly from that in English because they occur at the end of the sentence (verb-end position) in many clause types. Positions of pronouns can also diverge due to a more flexible word order in German. It is difficult for an HMM-based aligner like eflomal to model such high-distortion alignments, a property that has been found by prior work as well (Ho and Yvon, 2019). In contrast, mBERT(Argmax) does not use distortion information, so high distortion is not a problem for it.

Figure 7 gives an example for auxiliaries. The gold alignment ("has" – "hat") is correctly identified by mBERT (solid line). Eflomal generates an incorrect alignment ("time" – "hat"): the two words have about the same relative position, indicating that distortion minimization is the main reason for this incorrect alignment. Analyzing all auxiliary alignment edges, the average absolute value of the distance between aligned words is 2.72 for eflomal and 3.22 for mBERT. This indicates that eflomal is more reluctant than mBERT to generate high-distortion alignments and thus loses accuracy.

5 Related Work

Brown et al. (1993) introduced the IBM models, the best known statistical word aligners. More recent aligners, often based on IBM models, include fast-align (Dyer et al., 2013), Giza++ (Och and Ney, 2003) and eflomal (Östling and Tiedemann, 2016). (Östling, 2015a) showed that Bayesian Alignment Models perform well. Neural network based extensions of these models have been considered (Ayan et al., 2005; Ho and Yvon, 2019). All of these models are trained on parallel text. Our method instead aligns based on embeddings that are induced from monolingual data only. We compare with prior methods and observe comparable performance.

Prior work on using learned representations for alignment includes (Smadja et al., 1996; Och and Ney, 2003) (Dice coefficient), (Jalili Sabet et al., 2016) (incorporation of embeddings into IBM models), (Legrand et al., 2016) (neural network alignment model) and (Pourdamghani et al., 2018) (embeddings are used to encourage words to align to similar words). Tamura et al. (2014) use recurrent neural networks to learn alignments. They use noise contrastive estimation to avoid supervision. Yang et al. (2013) train a neural network that uses pretrained word embeddings in the initial layer. All of this work requires parallel data. mBERT is used for word alignments in concurrent work: Libovický et al. (2019) use the high quality of mBERT alignments as evidence for the "language-neutrality" of mBERT. Nagata et al. (2020) phrase word alignment as crosslingual span prediction and finetune mBERT using gold alignments.

Attention in NMT (Bahdanau et al., 2014) is related to a notion of soft alignment, but often deviates from conventional word alignments (Ghader and Monz, 2017; Koehn and Knowles, 2017). One difference is that standard attention does not have access to the target word. To address this, Peter et al. (2017) tailor attention matrices to obtain higher quality alignments. Li et al. (2018)'s and Zenkel et al. (2019)'s models perform similarly to and Zenkel et al. (2020) outperform Giza++. Ding et al. (2019) propose better decoding algorithms to deduce word alignments from NMT predictions. Chen et al. (2016), Mi et al. (2016) and Garg et al. (2019) obtain alignments and translations in a multitask setup. Garg et al. (2019) find that operating at the subword level can be beneficial for alignment models. Li et al. (2019) propose two methods to extract alignments from NMT

models, however they do not outperform fast-align. Stengel-Eskin et al. (2019) compute similarity matrices of encoder-decoder representations that are leveraged for word alignments, together with supervised learning, which requires manually annotated alignment. We find our proposed methods to be competitive with these approaches. In contrast to our work, they all require parallel data.

6 Conclusion

We presented word aligners based on contextualized embeddings that outperform in four and match the performance of state-of-the-art aligners in two language pairs; e.g., for ENG-DEU contextualized embeddings achieve an alignment F_1 that is 5 percentage points higher than eflomal trained on 100k parallel sentences. Further, we showed that alignments from static embeddings can be a viable alternative to statistical aligner when few parallel training data is available. In contrast to all prior work our methods do not require parallel data for training at all. With our proposed methods and extensions such as Match, Itermax and Null it is easy to obtain higher precision or recall depending on the use case.

Future work includes modeling fertility explicitly and investigating how to incorporate parallel data into the proposed methods.

Acknowledgments

We gratefully acknowledge funding through a Zentrum Digitalisierung.Bayern fellowship awarded to the second author. This work was supported by the European Research Council (# 740516). We thank Matthias Huck, Jindřich Libovický, Alex Fraser and the anonymous reviewers for interesting discussions and valuable comments. Thanks to Jindřich for pointing out that mBERT can align mixed-language sentences as shown in Figure 1.

References

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, Belgium, Brussels. Association for Computational Linguistics.

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, Berlin, Germany. Association for Computational Linguistics.

Tamer Alkhouli and Hermann Ney. 2017. Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

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

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)*, Melbourne, Australia. Association for Computational Linguistics.

Ehsaneddin Asgari and Hinrich Schütze. 2017. Past, present, future: A computational investigation of the typology of tense in 1000 languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics.

Necip Fazil Ayan, Bonnie J. Dorr, and Christof Monz. 2005. NeurAlign: Combining word alignments using neural networks. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

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

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).

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *AMTA 2016*.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California. Association for Computational Linguistics.

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*, Online. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of the Sixth International Conference on Learning Representations*.

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)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, Florence, Italy. Association for Computational Linguistics.

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)*, Melbourne, Australia. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia. Association for Computational Linguistics.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, Dublin, Ireland. European Association for Machine Translation.

Zvi Galil. 1986. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)*, 18(1).

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. 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)*, Hong Kong, China. Association for Computational Linguistics.

Ulrich Germann. 2001. Aligned Hansards of the 36th parliament of Canada.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *Thirtieth AAAI Conference on Artificial Intelligence*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Anh Khoa Ngo Ho and François Yvon. 2019. Neural baselines for word alignment. In *Proceedings of the 16th International Workshop on Spoken Language Translation*.

Matthias Huck, Diana Dutka, and Alexander Fraser. 2019. Cross-lingual annotation projection is effective for neural part-of-speech tagging. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 223–233, Ann Arbor, Michigan. Association for Computational Linguistics.

Masoud Jalili Sabet, Heshaam Faili, and Gholamreza Haffari. 2016. Improving word alignment of rare words with word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan. The COLING 2016 Organizing Committee.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit*, volume 5.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and

David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *International Workshop on Spoken Language Translation (IWSLT) 2005*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, Vancouver. Association for Computational Linguistics.

Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2).

Joël Legrand, Michael Auli, and Ronan Collobert. 2016. Neural network-based word alignment through score aggregation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, Berlin, Germany. Association for Computational Linguistics.

William D. Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Xintong Li, Lemao Liu, Zhaopeng Tu, Shuming Shi, and Max Meng. 2018. Target foresight based attention for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana. Association for Computational Linguistics.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual BERT? *arXiv preprint arXiv:1911.03310*.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan. The COLING 2016 Organizing Committee.

David Mareček. 2008. Automatic alignment of tectogrammatical trees from Czech-English parallel corpus. Master's thesis, Charles University, MFF UK.

Anthony McEnery, Paul Baker, Rob Gaizauskas, and Hamish Cunningham. 2000. Emille: Building a corpus of South Asian languages. *VIVEK-BOMBAY-*, 13(3).

I. Dan Melamed. 2000. Models of translation equivalence among words. *Computational Linguistics*, 26(2).

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas. Association for Computational Linguistics.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*.

Masaaki Nagata, Chousa Katsuki, and Masaaki Nishino. 2020. A supervised word alignment method based on cross-language span prediction using multilingual BERT. *arXiv preprint arXiv:2004.14516*.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Robert Östling. 2015a. *Bayesian models for multilingual word alignment*. Ph.D. thesis, Department of Linguistics, Stockholm University.

Robert Östling. 2015b. Word order typology through multilingual word alignment. 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)*, Beijing, China. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1).

Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36.

Jan-Thorsten Peter, Arne Nix, and Hermann Ney. 2017. Generating alignments using target foresight in attention-based neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1).

Mohammad Taher Pilevar, Heshaam Faili, and Abdol Hamid Pilevar. 2011. TEP: Tehran English-Persian parallel corpus. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer.

Nima Pourdamghani, Marjan Ghazvininejad, and Kevin Knight. 2018. Using word vectors to improve word alignments for low resource machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana. Association for Computational Linguistics.

Ofir Press and Noah A Smith. 2018. You may not need attention. *arXiv preprint arXiv:1810.13409*.

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*, Online. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

Frank Smadja, Kathleen R. McKeown, and Vasileios Hatzivassiloglou. 1996. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1).

Elias Stengel-Eskin, Tzu-ray Su, Matt Post, and Benjamin Van Durme. 2019. A discriminative neural model for cross-lingual word alignment. 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)*, Hong Kong, China. Association for Computational Linguistics.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2014. Recurrent neural networks for word alignment model. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland. Association for Computational Linguistics.

Leila Tavakoli and Heshaam Faili. 2014. Phrase alignments in parallel corpus using bootstrapping approach. *International Journal of Information & Communication Technology Research*, 6(3).

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word alignment modeling with context dependent deep neural network. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria. 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*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

A Additional Non-central Results

A.1 Comparison with Prior Work

A more detailed version of Table 2 from the main paper that includes precision and recall and results on Itermax can be found in Table 7.

A.2 Rare Words

Figure 8 shows the same as Figure 6 from the main paper but now with a reference corpus of 100k/1000k instead of 1920k parallel sentences. The main takeaways are similar.

A.3 Symmetrization

For asymmetric alignments different symmetrization methods exist. Dyer et al. (2013) provide an overview and implementation (fast-align) for these methods, which we use. We compare intersection and grow-diag-final-and (GDFA) in Table 9. In terms of F1, GDFA performs better (Intersection wins four times, GDFA eleven times, three ties). As expected, Intersection yields higher precision while GDFA yields higher recall. Thus intersection is preferable for tasks like annotation projection,

		Method	ENG-CES Prec.	Rec.	F_1	AER	ENG-DEU Prec.	Rec.	F_1	AER	ENG-FAS Prec.	Rec.	F_1	AER	ENG-FRA Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER	ENG-RON Prec.	Rec.	F_1	AER
Prior Work		(Östling, 2015a) Bayesian													.96	.92	**.94**	**.06**	.85	.43	.57	.43	.91	.61	**.73**	**.27**
		(Östling, 2015a) Giza++													**.98**	.87	.92	.07	.63	.44	.51	.49	.85	.63	.72	.28
		(Legrand et al., 2016) Ensemble Method	.79	.83	.81	.16									.59	.90	.71	.10								
		(Östling and Tiedemann, 2016) efmaral															.93	.08			.53	.47			.72	.28
		(Östling and Tiedemann, 2016) fast-align															.86	.15			.33	.67			.68	.33
		(Zenkel et al., 2019) Giza++								.21								**.06**								.28
		(Garg et al., 2019) Multitask								.20								.08								
Baselines	Word	fast-align/IBM2	.71	.81	.76	.25	.70	.73	.71	.29	.60	.54	.57	.43	.81	.93	.86	.15	.34	.33	.34	.66	.69	**.67**	.68	.33
		Giza++/IBM4	.71	.79	.75	.26	.79	.75	.77	.23	.55	.48	.51	.49	.90	.95	.92	.09	.47	.43	.45	.55	.74	.64	.69	.31
		eflomal	.84	.86	.85	.15	.80	.75	.77	.23	.68	.55	.61	.39	.91	.94	.93	.08	.61	.44	.51	.49	.81	.63	.71	.29
	Subword	fast-align/IBM2	.72	.84	.78	.23	.67	.74	.71	.30	.60	.56	.58	.42	.80	.92	.85	.16	.39	.37	.38	.62	.69	**.67**	.68	.32
		Giza++/IBM4	.79	.86	.82	.18	.78	.78	.78	.22	.58	.56	.57	.43	.89	.95	.92	.09	.52	.44	.48	.52	.74	.64	.69	.32
		eflomal	.80	.88	.84	.17	.74	.78	.76	.24	.66	.60	.63	.37	.88	.95	.91	.09	.58	.47	.52	.48	.78	**.67**	.72	.28
This Work	Word	fastText - Itermax	.74	.69	.72	.29	.69	.56	.62	.38	.63	.45	.53	.48	.74	.78	.76	.24	.63	.42	.51	.49	.64	.40	.50	.51
		mBERT[8] - Itermax	.87	.87	**.87**	.14	.85	.77	**.81**	**.19**	.80	.63	.70	.30	.91	.95	.93	.08	.75	.47	.58	.43	.82	.58	.68	.32
		XLM-R[8] - Itermax	.89	.85	**.87**	**.13**	.86	.73	.79	.21	.84	.63	**.72**	**.28**	.91	.93	.92	.08	.79	.49	.61	**.39**	.87	.61	.71	.29
		fastText - Argmax	.86	.59	.70	.30	.81	.48	.60	.40	.75	.38	.50	.50	.85	.71	.77	.22	.75	.36	.49	.52	.77	.34	.47	.53
		mBERT[8] - Argmax	.95	.80	**.87**	**.13**	.92	.69	.79	.21	.88	.54	.67	.33	.97	.91	**.94**	**.06**	.84	.39	.54	.47	.90	.50	.64	.36
		XLM-R[8] - Argmax	**.96**	.80	**.87**	**.13**	**.93**	.68	.79	.22	**.91**	.57	.70	.30	.96	.91	.93	**.06**	**.88**	.45	.59	.41	**.94**	.56	.70	.30
	Subword	fastText - Itermax	.61	.57	.59	.41	.63	.54	.58	.42	.20	.07	.11	.90	.70	.76	.73	.28	.14	.05	.07	.93	.56	.38	.45	.55
		mBERT[8] - Itermax	.84	**.89**	.86	.14	.83	**.80**	**.81**	**.19**	.76	.65	.70	.30	.91	**.96**	.93	.08	.71	.49	.58	.42	.79	.62	.69	.31
		XLM-R[8] - Itermax	.84	**.89**	.86	.14	.83	.78	.80	.20	.79	**.67**	**.72**	**.28**	.89	.94	.92	.09	.75	**.52**	**.62**	**.39**	.83	.64	.72	.28
		fastText - Argmax	.72	.48	.58	.42	.75	.45	.56	.44	.27	.06	.09	.91	.80	.67	.73	.26	.14	.02	.04	.96	.67	.31	.43	.58
		mBERT[8] - Argmax	.92	.81	.86	.14	.92	.72	**.81**	**.19**	.85	.56	.67	.33	.96	.92	**.94**	**.06**	.81	.41	.55	.45	.88	.51	.65	.35
		XLM-R[8] - Argmax	.92	.83	**.87**	**.13**	.92	.72	**.81**	**.19**	.87	.59	.71	.30	.95	.91	.93	.07	.86	.47	.61	**.39**	.91	.59	.71	.29

Table 7: Comparison of word and subword levels. Best overall result per column in bold.

Emb.	Method	ENG-DEU Prec.	Rec.	F_1	AER	ENG-CES Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER
fastText	Argmax	.75	.45	.56	.44	.72	.48	.58	.42	.14	.02	.04	.96
	+Dist	**.79**	**.51**	**.62**	**.38**	**.77**	**.58**	**.66**	**.34**	**.16**	**.04**	**.06**	**.94**
	+Null	.76	.43	.55	.45	.74	.47	.57	.42	.14	.02	.04	.96
	Itermax	.63	.54	.58	.42	.61	.57	.59	.41	.14	.05	.07	.93
	+Dist	**.67**	**.60**	**.64**	**.36**	**.63**	**.66**	**.65**	**.36**	**.15**	**.07**	**.09**	**.91**
	+Null	.64	.52	.57	.43	.62	.56	.59	.41	.14	.04	.07	.93
	Match	.51	.58	.54	.46	.44	.61	.52	.49	**.10**	.08	**.09**	**.91**
	+Dist	**.59**	**.66**	**.62**	**.38**	**.54**	**.71**	**.61**	**.39**	**.10**	**.09**	**.09**	**.91**
	+Null	.52	.57	.54	.46	.46	.60	.52	.48	**.10**	.08	**.09**	**.91**
mBERT[8]	Argmax	.92	**.72**	**.81**	**.19**	**.92**	**.81**	**.86**	**.14**	.81	**.41**	**.55**	**.45**
	+Dist	.90	.70	.79	.21	.91	.80	.85	.15	.65	.30	.41	.59
	+Null	**.93**	.70	.80	.20	**.92**	.78	.85	.15	**.82**	.40	.54	.47
	Itermax	.83	**.80**	**.81**	**.19**	.84	**.89**	**.86**	**.14**	.71	**.49**	**.58**	**.42**
	+Dist	.81	.77	.79	.21	.82	.87	.84	.16	.53	.35	.42	.58
	+Null	**.85**	.77	**.81**	.20	**.84**	.86	.85	.15	**.72**	.47	.57	.43
	Match	.75	**.80**	**.78**	**.23**	.76	**.90**	**.82**	**.18**	.64	**.52**	**.58**	**.43**
	+Dist	.72	.77	.75	.26	.74	.88	.80	.20	.45	.37	.40	.60
	+Null	**.77**	.78	**.78**	**.23**	**.77**	.88	**.82**	.19	**.65**	.51	.57	**.43**

Table 8: Comparison of methods for inducing alignments from similarity matrices. All results are subword-level. Best result per embedding type across columns in bold.

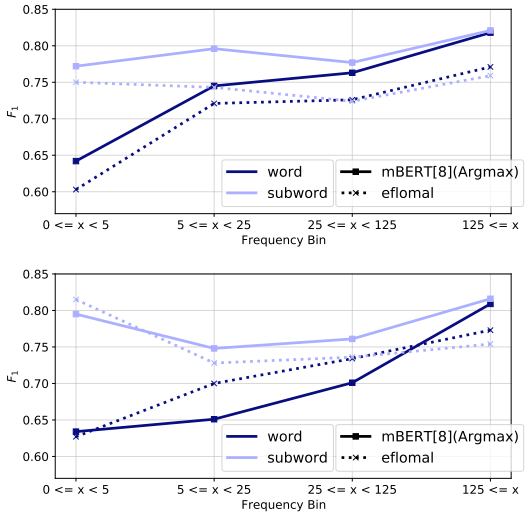


Figure 8: Results for different frequency bins. An edge in S, P, or A is attributed to exactly one bin based on the minimum frequency of the involved words (denoted by x). Top: Eflomal trained and frequencies computed on 100k parallel sentences. Bottom: 1000k parallel sentences.

whereas GDFA is typically used in statistical machine translation.

A.4 Alignment Examples for Different Methods

We show examples in Figure 10, Figure 11, Figure 12, and Figure 13. They provide an overview how the methods actually affect results.

B Hyperparameters

B.1 Overview

We provide a list of customized hyperparameters used in our computations in Table 10. There are three options how we came up with the hyperparameters: a) We simply used default values of 3rd party software. b) We chose an arbitrary value.

Method	Symm.	ENG-CES Prec.	Rec.	F_1	AER	ENG-DEU Prec.	Rec.	F_1	AER	ENG-FAS Prec.	Rec.	F_1	AER	ENG-FRA Prec.	Rec.	F_1	AER	ENG-HIN Prec.	Rec.	F_1	AER	ENG-RON Prec.	Rec.	F_1	AER
eflomal	Inters.	**.95**	.79	**.86**	**.14**	**.91**	.66	.76	.24	**.88**	.43	.58	.42	**.96**	.90	**.93**	**.07**	**.81**	.37	**.51**	**.49**	**.91**	.56	.70	.31
	GDFA	.84	**.86**	.85	.15	.80	**.75**	**.77**	**.23**	.68	**.55**	**.61**	**.39**	.91	**.94**	**.93**	.08	.61	**.44**	**.51**	**.49**	.81	**.63**	**.71**	**.29**
fast-align	Inters.	**.89**	.69	**.78**	**.22**	**.87**	.60	**.71**	**.29**	**.78**	.43	.55	.45	**.93**	.84	**.88**	**.11**	**.55**	.22	.31	.69	**.89**	.50	.64	.36
	GDFA	.71	**.81**	.76	.25	.70	**.73**	**.71**	**.29**	.60	**.54**	**.57**	**.43**	.81	**.93**	.86	.15	.34	**.33**	**.34**	**.66**	.69	**.67**	**.68**	**.33**
GIZA++	Inters.	**.95**	.60	.74	**.26**	**.92**	.62	.74	.26	**.89**	.26	.40	.60	**.97**	.89	**.93**	**.06**	**.82**	.25	.38	.62	**.95**	.47	.63	.37
	GDFA	.71	**.79**	**.75**	**.26**	.79	**.75**	**.77**	**.23**	.55	**.48**	**.51**	**.49**	.90	**.95**	.92	.09	.47	**.43**	**.45**	**.55**	.74	**.64**	**.69**	**.31**

Table 9: Comparison of symmetrization methods at the word level. Best result across rows per method in bold.

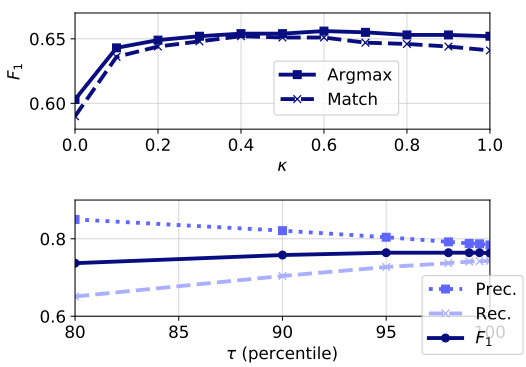


Figure 9: Top: F1 for ENG-DEU with fastText at word-level for different values of κ. Bottom: Performance for ENG-DEU with mBERT[8] (Match) at word-level when setting the value of τ to different percentiles. τ can be used for trading precision against recall. F_1 remains stable although it decreases slightly when assigning τ the value of a smaller percentile (e.g., 80).

Usually we fell back to well-established and rather conventional values (e.g., embedding dimension 300 for static embeddings). c) We defined a reasonable but arbitrary range, out of which we selected the best value using grid search. Table 10 lists the final values we used as well as how we came up with the specific value. For option c) the corresponding analyses are in Figure 4 and Table 3 in the main paper as well as in §B.2 in this supplementary material.

B.2 Null and Distortion Extensions

In Figure 9 we plot the performance for different values of κ. We observe that introducing distortion indeed helps (i.e., $\kappa > 0$) but the actual value is not decisive for performance. This is rather intuitive, as a small adjustment to the similarities is sufficient while larger adjustments do not necessarily change the argmax or the optimal point in the matching algorithm. We choose $\kappa = 0.5$.

For τ in null-word extension, we plot precision, recall and F_1 in Figure 9 when assigning τ different percentile values. Note that values for τ depend on the similarity distribution of all aligned edges. As expected, when using the 100th percentile no edges are removed and thus the performance is not changed compared to not having a null-word extension. When decreasing the value of τ the precision increases and recall goes down, while F_1 remains stable. We use the 95th percentile for τ .

C Reproducibility Information

C.1 Computing Infrastructures, Runtimes, Number of Parameters

We did all computations 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.

Runtimes for aligning 500 parallel sentences on ENG-DEU are reported in Table 12. mBERT and XLM-R computations are done on the GPU. Note that fast-align, GIZA++ and eflomal usually need to be trained on much more parallel data to achieve better performance: this increases their runtime.

All our proposed methods are **parameter-free**. If we consider the parameters of the pretrained language models and pretrained embeddings then fastText has around 1 billion parameters (up to 500k words per language, 7 languages and embedding dimension 300), mBERT has 172 million, XLM-R 270 million parameters.

Method	Runtime[s]
fast-align	4
GIZA++	18
eflomal	5
mBERT[8] - Argmax	15
XLM-R[8] - Argmax	22

Table 12: Runtime (average across 5 runs) in seconds for each method to align 500 parallel sentences.

C.2 Data

Table 11 provides download links to all data used.

System	Parameter	Value
fastText	Version	0.9.1
	Code URL	https://github.com/facebookresearch/fastText/archive/v0.9.1.zip
	Downloaded on	11.11.2019
	Embedding Dimension	300
mBERT,XLM-R	Code: Huggingface Transformer	Version 2.3.1
	Maximum Sequence Length	128
fastalign	Code URL	https://github.com/clab/fast_align
	Git Hash	7c2bbca3d5d61ba4b0f634f098c4fcf63c1373e1
	Flags	-d -o -v
eflomal	Code URL	https://github.com/robertostling/eflomal
	Git Hash	9ef1ace1929c7687a4817ec6f75f47ee684f9aff
	Flags	–model 3
GIZA++	Code URL	http://web.archive.org/web/20100221051856/http://code.google.com/p/giza-pp
	Version	1.0.3
	Iterations	5 iter. HMM, 5 iter. Model 1, 5 iter. Model3, 5 iter. Model 4 (DEFAULT)
	p0	0.98
Vecmap	Code URL	https://github.com/artetxem/vecmap.git
	Git Hash	b82246f6c249633039f67fa6156e51d852bd73a3
	Manual Vocabulary Cutoff	500000
Distortion Ext.	κ	0.5 (chosen ouf of $[0.0, 0.1, \ldots, 1.0]$ by grid search, criterion: F_1)
Null Extension	τ	95th percentile of similarity distribution of aligned edges (chosen out of [80, 90, 95, 98, 99, 99.5] by grid search, criterion: F_1)
Argmax	Layer	8 (for mBERT and XLM-R, chosen out of $[0, 1, \ldots, 12]$ by grid search, criterion: F_1)
Vecmap	α	0.9 (chosen out of [0.9, 0.95, 1] by grid search, criterion: F_1)
	Iterations $n_{\max}$	2 (chosen out of [1,2,3] by grid search, criterion: F_1)

Table 10: Overview on hyperparameters. We only list parameters where we do **not** use default values. Shown are the values which we use unless specifically indicated otherwise.

Lang.	Name	Description	Link
ENG-CES	(Mareček, 2008)	Gold Alignment	http://ufal.mff.cuni.cz/czech-english-manual-word-alignment
ENG-DEU	EuroParl-based	Gold Alignment	www-i6.informatik.rwth-aachen.de/goldAlignment/
ENG-FAS	(Tavakoli and Faili, 2014)	Gold Alignment	http://eceold.ut.ac.ir/en/node/940
ENG-FRA	WPT2003, (Och and Ney, 2000),	Gold Alignment	http://web.eecs.umich.edu/ mihalcea/wpt/
ENG-HIN	WPT2005	Gold Alignment	http://web.eecs.umich.edu/ mihalcea/wpt05/
ENG-RON	WPT2005 (Mihalcea and Pedersen, 2003)	Gold Alignment	http://web.eecs.umich.edu/ mihalcea/wpt05/
ENG-CES	EuroParl (Koehn, 2005)	Parallel Data	https://www.statmt.org/europarl/
ENG-DEU	EuroParl (Koehn, 2005)	Parallel Data	https://www.statmt.org/europarl/
ENG-DEU	ParaCrawl	Parallel Data	https://paracrawl.eu/
ENG-FAS	TEP (Pilevar et al., 2011)	Parallel Data	http://opus.nlpl.eu/TEP.php
ENG-FRA	Hansards (Germann, 2001)	Parallel Data	https://www.isi.edu/natural-language/download/hansard/index.html
ENG-HIN	Emille (McEnery et al., 2000)	Parallel Data	http://web.eecs.umich.edu/mihalcea/wpt05/
ENG-RON	Constitution, Newspaper	Parallel Data	http://web.eecs.umich.edu/ mihalcea/wpt05/
All langs.	Wikipedia (downloaded October 2019)	Monolingual Text	download.wikimedia.org/[X]wiki/latest/[X]wiki-latest-pages-articles.xml.bz2

Table 11: Overview of datasets. “Lang.” uses ISO 639-3 language codes.

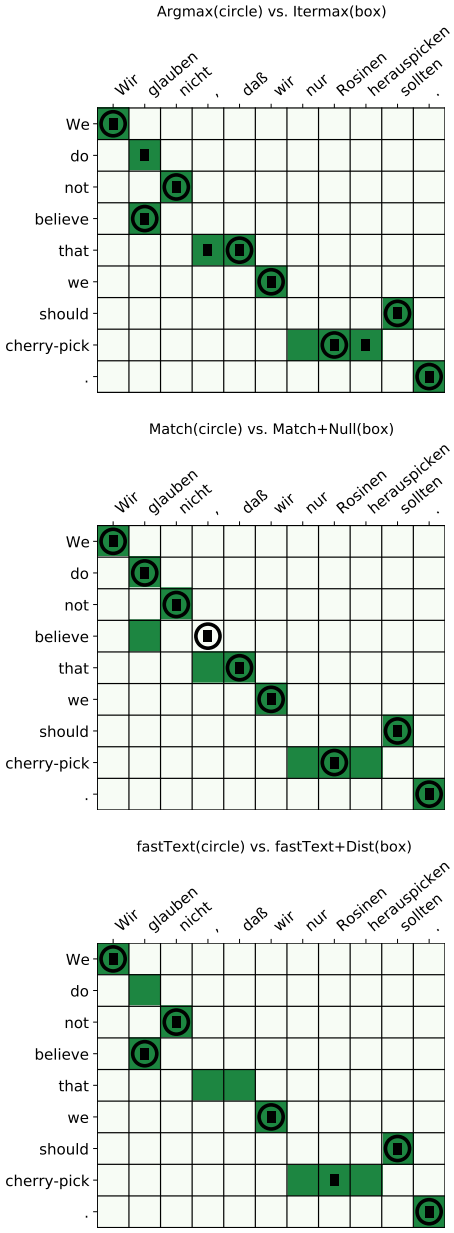


Figure 10: Comparison of alignment methods. Dark/light green: sure/possible edges in the gold standard. Circles are alignments from the first mentioned method in the subfigure title, boxes alignments from the second method.

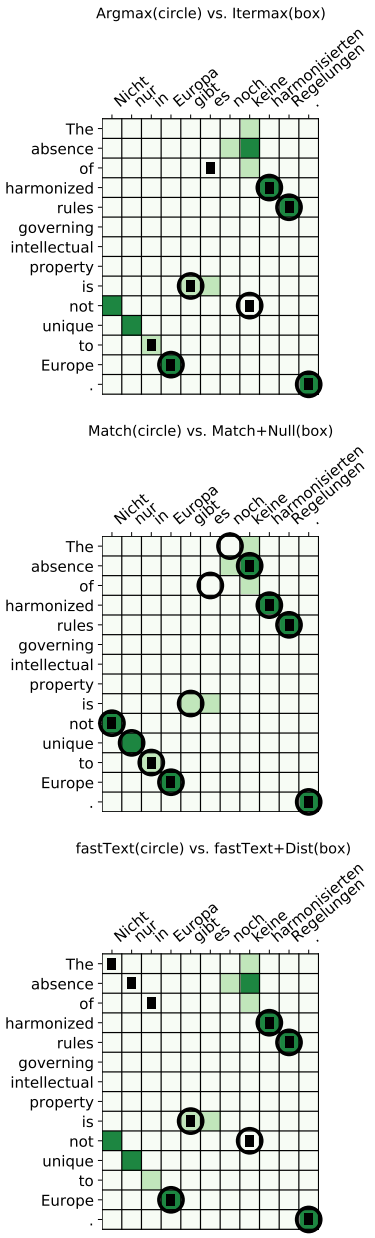


Figure 11: More examples.

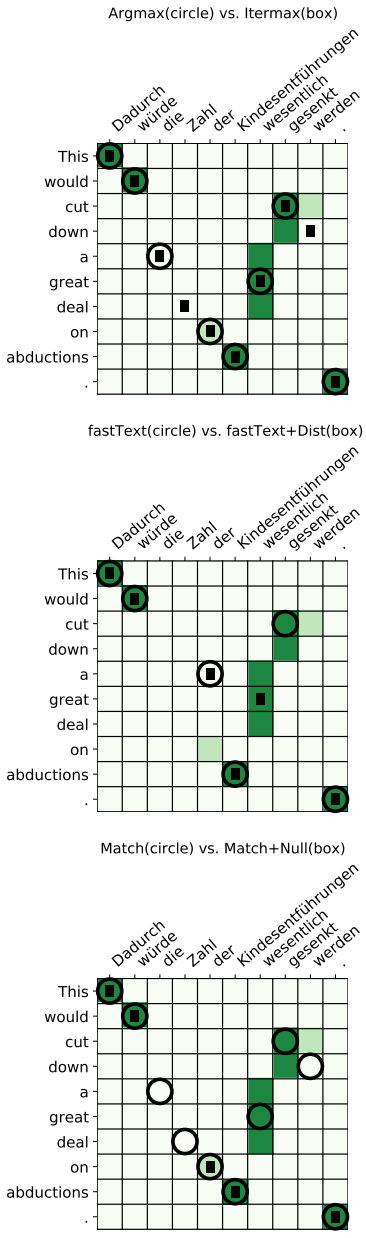


Figure 12: More examples.

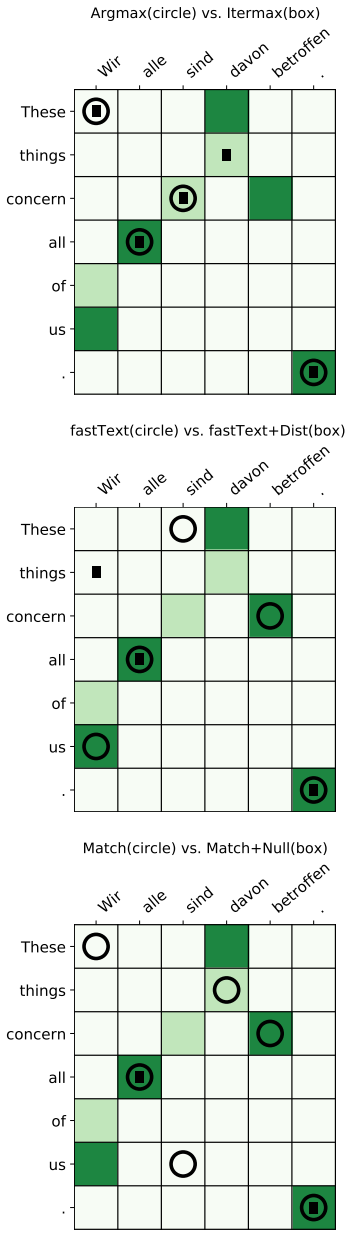


Figure 13: More examples.

Chapter 6

Identifying Elements Essential for BERT's Multilinguality

Identifying Elements Essential for BERT's Multilinguality

Philipp Dufter, Hinrich Schütze
Center for Information and Language Processing (CIS), LMU Munich, Germany
philipp@cis.lmu.de

Abstract

It has been shown that multilingual BERT (mBERT) yields high quality multilingual representations and enables effective zero-shot transfer. This is surprising given that mBERT does not use any crosslingual signal during training. While recent literature has studied this phenomenon, the reasons for the multilinguality are still somewhat obscure. We aim to identify architectural properties of BERT and linguistic properties of languages that are necessary for BERT to become multilingual. To allow for fast experimentation we propose an efficient setup with small BERT models trained on a mix of synthetic and natural data. Overall, we identify four architectural and two linguistic elements that influence multilinguality. Based on our insights, we experiment with a multilingual pretraining setup that modifies the masking strategy using VecMap, i.e., unsupervised embedding alignment. Experiments on XNLI with three languages indicate that our findings transfer from our small setup to larger scale settings.

1 Introduction

Multilingual models, i.e., models capable of processing more than one language with comparable performance, are central to natural language processing. They are useful as fewer models need to be maintained to serve many languages, resource requirements are reduced, and low- and mid-resource languages can benefit from crosslingual transfer. Further, multilingual models are useful in machine translation, zero-shot task transfer and typological research. There is a clear need for multilingual models for the world's 7000+ languages.

With the rise of static word embeddings, many multilingual embedding algorithms have been proposed (Mikolov et al., 2013; Hermann and Blunsom, 2014; Faruqui and Dyer, 2014); for a survey see (Ruder et al., 2019). Pretrained language models (Peters et al., 2018; Howard and Ruder, 2018; Devlin et al., 2019) have high performance across tasks, outperforming static word embeddings. A simple multilingual model is multilingual BERT[1] (*mBERT*). It is a BERT-Base model (Devlin et al., 2019) trained on the 104 largest Wikipedias with a shared subword vocabulary. There is no additional crosslingual signal. Still, mBERT yields high-quality multilingual representations (Pires et al., 2019; Wu and Dredze, 2019; Hu et al., 2020).

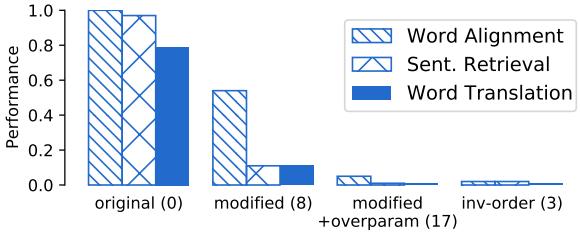


Figure 1: Multilinguality in our BERT model (0) is harmed by three architectural modifications: lang-pos, shift-special, no-random (8); see §2.3 for definitions. Together with overparameterization almost no multilinguality is left (17). Pairing a language with its inversion (i.e., inverted word order) destroys multilinguality as well (3). Having parallel training corpora is helpful for multilinguality (not shown). Results are for embeddings from layer 8.

The exact reason for mBERT's multilinguality is – to the best of our knowledge – still debated. K et al. (2020) provide an extensive study and conclude that a shared vocabulary is not necessary, but that the model needs to be deep and languages need to share a similar "structure". Artetxe et al. (2020) show that neither a shared vocabulary nor joint pretraining is required for BERT to be multilingual. Conneau et al. (2020b) find that BERT models across languages can be easily aligned and

[1] https://github.com/google-research/bert/blob/master/multilingual.md

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 4423–4437,
November 16–20, 2020.

that a necessary requirement for achieving multilinguality are shared parameters in the top layers. This work continues this line of research. We find indications that six elements influence the multilinguality of BERT. Figure 1 summarizes our main findings.

1.1 Contributions

- Training BERT models consumes tremendous resources. We propose an experimental setup that allows for fast experimentation.
- We hypothesize that BERT is multilingual because of a limited number of parameters. By forcing the model to use its parameters efficiently, it exploits common structures by aligning representations across languages. We provide experimental evidence that the number of parameters and training duration is interlinked with multilinguality and an indication that generalization and multilinguality might be conflicting goals.
- We show that shared special tokens, shared position embeddings and the common masking strategy to replace masked tokens with random words contribute to multilinguality. This is in line with findings from (Conneau et al., 2020b).
- We show that having identical structure across languages, but an inverted word order in one language destroys multilinguality. Similarly having shared position embeddings contributes to multilinguality. We thus hypothesize that word order across languages is an important ingredient for multilingual models.
- Using these insights we perform initial experiments to create a model with higher degree of multilinguality.
- We conduct experiments on Wikipedia and evaluate on XNLI to show that our findings transfer to larger scale settings.

Our code is publicly available.[2]

2 Setup and Hypotheses

2.1 Setup

We aim at having a setup that allows for gaining insights quickly when investigating multilinguality.

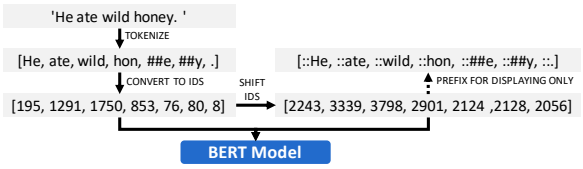


Figure 2: Creating a Fake-English sentence by adding a shift of 2048 to token indices.

Our assumption is that these insights are transferable to a larger scale real world setup. We verify this assumption in §5.

Languages. K et al. (2020) propose to consider English and Fake-English, a language that is created by shifting unicode points by a large constant. Fake-English in their case has the exact same linguistic properties as English, but is represented by different unicode points. We follow a similar approach, but instead of shifting unicode points we simply shift token indices after tokenization by a constant; shifted tokens are prefixed by "::" and added to the vocabulary. See Figure 2 for an example. While shifting indices and unicode code points have similar effects, we chose shifting indices as we find it somewhat cleaner.[3]

Data. For our setup, aimed at supporting fast experimentation, a small corpus with limited vocabulary is desirable. As training data we use the English Easy-to-Read version of the Parallel Bible Corpus (Mayer and Cysouw, 2014) that contains the New Testament. The corpus is structured into verses and is word-tokenized. We sentence-split verses using NLTK (Loper and Bird, 2002). The final corpus has 17k sentences, 228k words, a vocabulary size of 4449 and 71 distinct characters. The median sentence length is 12 words. By creating a Fake-English version of this corpus we get a shifted replica and thus a sentence-parallel corpus.

As development data we apply the same procedure to the first 10k sentences of the Old Testament of the English King James Bible. All our evaluations are performed on development data, except for word translation and when indicated explicitly.

Vocabulary. We create a vocabulary of size 2048 from the Easy-to-Read Bible with the wordpiece tokenizer (Schuster and Nakajima, 2012).[4]

[2] https://github.com/pdufter/minimult

[3] For example, the BERT tokenizer treats some punctuation as special symbols (e.g., "dry-cleaning" is tokenized as ["dry", "-", "##cleaning"], not as ["dry", "##-", "##cleaning"]). When using a unicode shift, tokenizations of English and Fake-English can differ.

[4] https://github.com/huggingface/tokenizers

Using the same vocabulary for English and Fake-English yields a final vocabulary size of 4096.

Model. We use the BERT-Base architecture (Devlin et al., 2019), modified to achieve a smaller model: we divide hidden size, intermediate size of the feed forward layer and number of attention heads by 12; thus, hidden size is 64 and intermediate size 256. While this leaves us with a single attention head, K et al. (2020) found that the number of attention heads is important neither for overall performance nor for multilinguality. We call this smaller model *BERT-small*.

As a consistency check for our experiments we consider random embeddings in the form of a randomly initialized but untrained BERT model, referred to as ***"untrained"***.

Training Parameters. We mostly use the original training parameters as given in (Devlin et al., 2019). Learning rate and number of epochs was chosen to achieve reasonable perplexity on the training corpus (see supplementary for details). Unless indicated differently we use a batch size of 256, train for 100 epochs with AdamW (Loshchilov and Hutter, 2019) (learning rate 2e-3, weight decay .01, epsilon 1e-6), and use 50 warmup steps. We only use the masked-language-modeling objective, without next-sequence-prediction. With this setup we can train a single model in under 40 minutes on a single GPU (GeForce GTX 1080Ti). We run each experiment with five different seeds, and report mean and standard deviation.

2.2 Evaluation

We evaluate two properties of our trained language models: the *degree of multilinguality* and – as a consistency check – the *overall model fit* (i.e., is the trained language model of reasonable quality).

2.2.1 Multilinguality

We evaluate the degree of multilinguality with three tasks. Representations from different layers of BERT can be considered. We use layer 0 (uncontextualized) and layer 8 (contextualized). Several papers have found layer 8 to work well for monolingual and multilingual tasks (Tenney et al., 2019; Hewitt and Manning, 2019; Sabet et al., 2020). Note that representations from layer 0 include position and segment embeddings besides the token embeddings as well as layer normalization.

Word Alignment. Sabet et al. (2020) find that mBERT performs well on word alignment. By construction, we have a sentence-aligned corpus with English and Fake-English. The gold word alignment between two sentences is the identity alignment. We use this automatically created gold-alignment for evaluation.

To extract word alignments from BERT we use (Sabet et al., 2020)'s Argmax method. Consider the parallel sentences $s^{(\text{eng})}, s^{(\text{fake})}$, with length n. We extract d-dimensional wordpiece embeddings from the l-th layer of BERT to obtain embeddings $\mathcal{E}(s^{(k)}) \in \mathbb{R}^{n \times d}$ for $k \in \{\text{eng}, \text{fake}\}$. The similarity matrix $S \in [0,1]^{n \times n}$ is computed by $S_{ij} := \text{cosine-sim}\left(\mathcal{E}(s^{(\text{eng})})_i, \mathcal{E}(s^{(\text{fake})})_j\right)$. Two wordpieces i and j are aligned if

$$(i = \arg\max_l S_{l,j}) \wedge (j = \arg\max_l S_{i,l}).$$

The alignments are evaluated using precision, recall and F_1 as follows:

$$\text{p} = \frac{|P \cap G|}{|P|}, \ \text{r} = \frac{|P \cap G|}{|G|}, \ F_1 = \frac{2\ \text{p}\ \text{r}}{\text{p} + \text{r}},$$

where P is the set of predicted alignments and G the set of true alignment edges. We report F_1.

Sentence Retrieval is popular for evaluating crosslingual representations (e.g., (Artetxe and Schwenk, 2019; Libovický et al., 2019)). We obtain the embeddings $\mathcal{E}(s^{(k)})$ as before and compute a sentence embedding $e_s^{(k)}$ simply by averaging vectors across all tokens in a sentence (ignoring CLS and SEP tokens). Computing cosine similarities between English and Fake-English sentences yields the similarity matrix $R \in \mathbb{R}^{m \times m}$ where $R_{ij} = \text{cosine-sim}(e_i^{(\text{eng})}, e_j^{(\text{fake})})$ for m sentences.

Given an English query sentence $s_i^{(\text{eng})}$, we obtain the retrieved sentences in Fake-English by ranking them according to similarity. Since we can do the same with Fake-English as query language, we report the mean precision of these directions, computed as

$$\rho = \frac{1}{2m} \sum_{i=1}^{m} \mathbb{1}_{\arg\max_l R_{il}=i} + \mathbb{1}_{\arg\max_l R_{li}=i}.$$

We also evaluate **word translation.** Again, by construction we have a ground-truth bilingual dictionary of size 2048. We obtain word vectors by feeding each word in the vocabulary individually to BERT, in the form "[CLS] {token} [SEP]". We then evaluate word translation like sentence retrieval and denote the measure with τ.

Multilinguality Score. For an easier overview we compute a multilinguality score by averaging

retrieval and translation results across both layers. That is $\mu = 1/4(\tau_0 + \tau_8 + \rho_0 + \rho_8)$ where τ_k,ρ_k means representations from layer k have been used. We omit word alignment here as it is not a suitable measure to compare all models: with shared position embeddings, the task is almost trivial given that the gold alignment is the identity alignment.

2.2.2 Model Fit

MLM Perplexity. To verify that BERT was successfully trained we evaluate the models on perplexity (with base e) for training and development data. Perplexity is computed on 15% of randomly selected tokens that are replaced by "[MASK]". Given those randomly selected tokens in a text $w_1, \ldots, w_n$ and probabilities $p_{w_1}, \ldots, p_{w_n}$ that the correct token was predicted by the model, perplexity is calculated as $\exp(-1/n \sum_{k=1}^{n} \log(p_{w_k}))$.

2.3 Architectural Properties

Here we formulate hypotheses as to which architectural components contribute to multilinguality.

Overparameterization: *overparam*. If BERT is severely overparameterized the model should have enough capacity to model each language separately without creating a multilingual space. Conversely, if the number of parameters is small, the model has a need to use parameters efficiently. The model is likely to identify common structures among languages and model them together, thus creating a multilingual space.

To test this, we train a larger BERT model that has the same configuration as BERT-base (i.e., hidden size: 768, intermediate size: 3072, attention heads: 12) and is thus much larger than our standard configuration, BERT-small. Given our small training corpus and the small number of languages, we argue that BERT-base is overparameterized. For the overparameterized model we use learning rate 1e-4 (following (Devlin et al., 2019)).

Shared Special Tokens: *shift-special*. It has been found that a shared vocabulary is not essential for multilinguality (K et al., 2020; Artetxe et al., 2020; Conneau et al., 2020b). Similar to prior studies, in our setting each language has its own vocabulary, as we aim at breaking the multilinguality of BERT. However in prior studies, special tokens ([UNK], [CLS], [SEP], [MASK], [PAD]) are usually shared across languages. Shared special tokens may contribute to multilinguality because they are very frequent and could serve as "anchor points". To investigate this, we shift the special tokens with the same shift as applied to token indices.

	ENGLISH							FAKE-ENGLISH						
Tok.	195	1291	1750	853	76	80	8	2243	3339	3798	2901	2124	2128	2056
Pos.	1	2	3	4	5	6	7	129	130	131	132	133	134	135
Seg.	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Figure 3: lang-pos: input indices to BERT with language specific position and segment embeddings.

Shared Position Embeddings: *lang-pos*. Position and segment embeddings are usually shared across languages. We investigate their contribution to multilinguality by using language-specific position (*lang-pos*) and segment embeddings. For an example see Figure 3.

Random Word Replacement: *no-random*. The MLM task as proposed by Devlin et al. (2019) masks 15% of tokens in a sentence. These tokens are replaced with "[MASK]" in $p_{\text{[mask]}} = 80\%$, remain unchanged in $p_{\text{[id]}} = 10\%$ and are replaced with a random token of the vocabulary in $p_{\text{[rand]}} = 10\%$ of the cases. The randomly sampled token can come from any language resulting in Fake-English tokens to appear in English sentences and vice-versa. We hypothesize that this random replacement could contribute to multilinguality. We experiment with the setting $p = (0.8, 0.2, 0.0)$ where p denotes the triple $(p_{\text{[mask]}}, p_{\text{[id]}}, p_{\text{[rand]}})$.

2.4 Linguistic Properties

Inverted Word Order: *inv-order*. K et al. (2020) shuffled word order in sentences randomly and found that word order has some, but not a severe effect on multilinguality. They conclude that "structural similarity" across languages is important without further specifying this term. We investigate an extreme case: inversion. We invert each sentence in the Fake-English corpus: $[w_1, w_2, \ldots, w_n] \rightarrow [w_n, w_{n-1}, \ldots, w_1]$. Note that, apart from the reading order, all properties of the languages are preserved, including ngram statistics. Thus, the structural similarity of English and inverted Fake-English is arguably very high.

Comparability of Corpora: *no-parallel*. We hypothesize that the similarity of training corpora contributes to "structural similarity": if we train on a parallel corpus we expect the language structures to be more similar than when we train on two independent corpora, potentially from different domains. For mBERT, Wikipedias across languages are in the same domain, share some articles and thus are comparable, yet not parallel. To test our

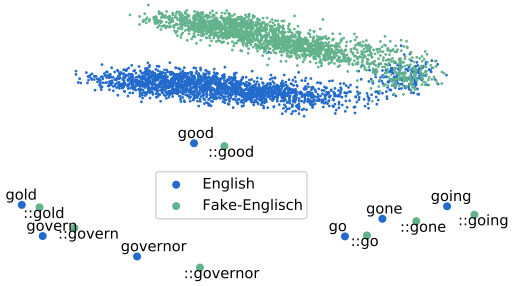


Figure 4: *Top*: PCA of the **token embeddings** from layer 0 of the original model (ID 0). The representations of the two languages clearly have a similar structure. *Bottom*: PCA of a sample of **token embeddings**. Corresponding tokens in English and Fake-English are nearest neighbors of each other or nearly so. This is quantitatively confirmed in Table 1.

hypothesis, we train on a non-parallel corpus. We create it by splitting the Bible into two halves, using one half for English and Fake-English each, thus avoiding any parallel sentences during training.

3 Results

3.1 Architectural Properties

Table 1 shows results. Each model has an associated ID that is consistent with the code. The original model (ID 0) shows a high degree of multilinguality. As mentioned, alignment is an easy task with shared position embeddings yielding $F_1 = 1.00$. Retrieval works better with contextualized representations on layer 8 (.97 vs. .16) whereas word translation works better on layer 0 (.88 vs. .79), as expected. Overall the embeddings seem to capture the similarity of English and Fake-English exceptionally well (see Figure 4 for a PCA of token embeddings). The untrained BERT models perform poorly (IDs 18, 19), except for word alignment with shared position embeddings.

When applying our **architectural modifications** (lang-pos, shift-special, no-random) individually we see medium to slight decreases in multilinguality (IDs 1, 2, 4). lang-pos has the largest negative impact. Apparently, applying just a single modification can be compensated by the model. Indeed, when using two modifications at a time (5–7) multilinguality goes down more, only with 7 there is still a high degree of multilinguality. With all three modifications (8) the degree of multilinguality is drastically lowered (μ .12 vs. .70).

We see that the language model quality (see columns MLM-Perpl.) is stable on train and dev across models (IDs 1–8) and does not deviate from

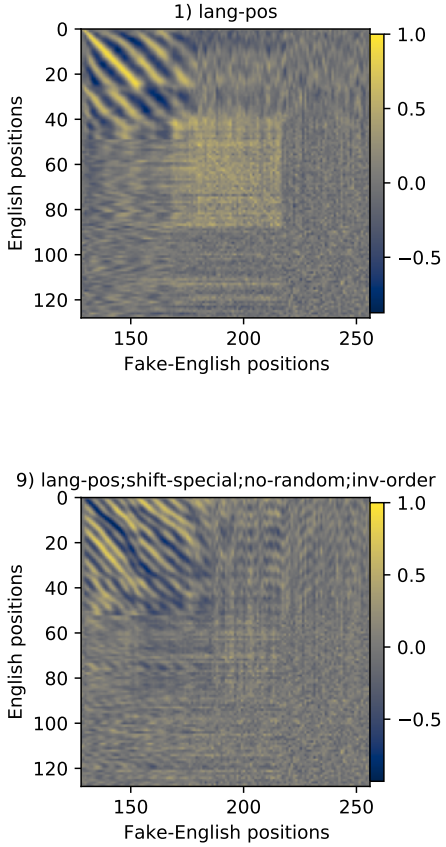


Figure 5: Cosine similarity matrices of position embeddings. The maximum length after tokenization in our experiments is 128. Position embedding IDs 0-127 are used by English, 128-255 by Fake-English.

original BERT (ID 0) by much.[5] Thus, we can conclude that each of the models has fitted the training data well and poor results on μ are not due to the fact that the architectural changes have hobbled BERT's language modeling performance.

The **overparameterized** model (ID 15) exhibits lower scores for word translation, but higher ones for retrieval and overall a lower multilinguality score (.58 vs. .70). However, when we add lang-pos (16) or apply all three architectural modifications (17), multilinguality drops to .01 and .00. This indicates that by decoupling languages with the proposed modifications (lang-pos, shift-special, no-random) and greatly increasing the number of parameters (overparam), it is possible to get a well-

[5]Perplexities on dev are high because the English of the King James Bible is quite different from that of the Easy-to-Read Bible. Our research question is: which modifications harm BERT's multilinguality without harming model fit (i.e., perplexity). The relative change of perplexities, not their absolute value is important in this context.

ID	Description	Mult.-score μ	Layer 0 Align. F_1	Layer 0 Retr. ρ	Layer 0 Trans. τ	Layer 8 Align. F_1	Layer 8 Retr. ρ	Layer 8 Trans. τ	MLM-Perpl. train	MLM-Perpl. dev
0	original	.70	1.00 $_{.00}$	.16 $_{.02}$	.88 $_{.02}$	1.00 $_{.00}$	.97 $_{.01}$	.79 $_{.03}$	9 $_{0.2}$	217 $_{7.8}$
1	lang-pos	.30	.87 $_{.05}$	.33 $_{.13}$	.40 $_{.09}$	.89 $_{.05}$	.39 $_{.15}$	.09 $_{.05}$	9 $_{0.1}$	216 $_{9.0}$
2	shift-special	.66	1.00 $_{.00}$	.15 $_{.02}$	.88 $_{.01}$	1.00 $_{.00}$	.97 $_{.02}$	.63 $_{.13}$	9 $_{0.1}$	227 $_{17.9}$
4	no-random	.68	1.00 $_{.00}$	.19 $_{.03}$	.87 $_{.02}$	1.00 $_{.00}$	.85 $_{.07}$	.82 $_{.04}$	9 $_{0.6}$	273 $_{7.7}$
5	lang-pos;shift-special	.20	.62 $_{.19}$	.22 $_{.19}$	.27 $_{.20}$	.72 $_{.22}$	.27 $_{.21}$	.05 $_{.04}$	10 $_{0.5}$	205 $_{7.6}$
6	lang-pos;no-random	.30	.91 $_{.04}$	.29 $_{.10}$	.36 $_{.12}$	.89 $_{.05}$	.32 $_{.15}$	.25 $_{.12}$	10 $_{0.4}$	271 $_{8.6}$
7	shift-special;no-random	.68	1.00 $_{.00}$	.21 $_{.03}$	.85 $_{.01}$	1.00 $_{.00}$	.89 $_{.06}$	.79 $_{.04}$	8 $_{0.3}$	259 $_{15.6}$
8	lang-pos;shift-special;no-random	.12	.46 $_{.26}$	.09 $_{.09}$	.18 $_{.22}$	.54 $_{.31}$	.11 $_{.11}$	.11 $_{.13}$	10 $_{0.6}$	254 $_{15.9}$
15	overparam	.58	1.00 $_{.00}$	.27 $_{.03}$	.63 $_{.05}$	1.00 $_{.00}$	.97 $_{.01}$	.47 $_{.06}$	2 $_{0.1}$	261 $_{4.5}$
16	lang-pos;overparam	.01	.25 $_{.10}$	.01 $_{.00}$	.01 $_{.00}$	.37 $_{.13}$	.01 $_{.00}$	.00 $_{.00}$	3 $_{0.0}$	254 $_{4.9}$
17	lang-pos;shift-special;no-random;overparam	.00	.05 $_{.02}$	.00 $_{.00}$	.00 $_{.00}$	.05 $_{.04}$	.00 $_{.00}$	.00 $_{.00}$	1 $_{0.0}$	307 $_{7.7}$
3	inv-order	.01	.02 $_{.00}$	.00 $_{.00}$	.01 $_{.00}$	.02 $_{.00}$	.01 $_{.01}$	.00 $_{.00}$	11 $_{0.3}$	209 $_{14.4}$
9	lang-pos;inv-order;shift-special;no-random	.00	.04 $_{.01}$	.00 $_{.00}$	.00 $_{.00}$	.03 $_{.01}$	.00 $_{.00}$	.00 $_{.00}$	10 $_{0.4}$	270 $_{20.1}$
18	untrained	.00	.97 $_{.01}$	.00 $_{.00}$	.00 $_{.00}$	.96 $_{.01}$	.00 $_{.00}$	.00 $_{.00}$	3484 $_{44.1}$	4128 $_{42.7}$
19	untrained;lang-pos	.00	.02 $_{.00}$	.00 $_{.00}$	.00 $_{.00}$	.02 $_{.00}$	.00 $_{.00}$	.00 $_{.00}$	3488 $_{41.4}$	4133 $_{50.3}$
30	knn-replace	.74	1.00 $_{.00}$	.31 $_{.08}$	.88 $_{.00}$	1.00 $_{.00}$	.97 $_{.01}$	.81 $_{.01}$	11 $_{0.3}$	225 $_{12.4}$

Table 1: Multilinguality and model fit for our models. Mean and standard deviation (subscript) across 5 different random seeds is shown. ID is a unique identifier for the model setting. To put perplexities into perspective: the pretrained mBERT has a perplexity of roughly 46 on train and dev. knn-replace is explained in §4.

ID	Description	μ	Layer 0 F_1	Layer 0 ρ	Layer 0 τ	Layer 8 F_1	Layer 8 ρ	Layer 8 τ	Perpl. train	Perpl. dev
0	original	.70	1.00	.16	.88	1.00	.97	.79	9	217
21	no-parallel	.25	.98	.06	.28	.98	.50	.15	14	383
21b	lang-pos;no-parallel	.07	.60	.10	.07	.73	.11	.02	16	456

Table 2: Results showing the effect of having a parallel vs. non-parallel training corpus.

performing language model (low perplexity) that is not multilingual. *Conversely, we can conclude that the four architectural properties together are necessary for BERT to be multilingual.*

3.2 Linguistic Properties

Inverting Fake-English (IDs 3, 9) breaks multilinguality almost completely – independently of any architectural modifications. Having a language with the exact same structure (same ngram statistics, vocabulary size etc.), only with inverted order, seems to block BERT from creating a multilingual space. Note that perplexity is almost the same. *We conclude that having a similar word order structure is necessary for BERT to create a multilingual space.* The fact that shared position embeddings are important for multilinguality supports this finding. Our hypothesis is that the drop in multilinguality with inverted word order comes from an incompatibility between word and position encodings: BERT needs to learn that the word at position 0 in English is similar to word at position n in Fake-English. However, n (the sentence length) varies from sentence to sentence. This suggests that relative position embeddings – rather than absolute position embeddings – might be beneficial for multilinguality across languages with high distortion.

To investigate this effect more, Figure 8 shows cosine similarities between position embeddings for models 1, 9. Position IDs 0-127 are for English, 128-255 for Fake-English. Despite language specific position embeddings, the embeddings exhibit a similar structure: in the top panel there is a clear yellow diagonal at the beginning, which weakens at the end. The bottom shows that for a model with inverted Fake-English the position embeddings live in different spaces: no diagonal is visible.

In the range 90–128 (a rare sentence length) the similarities look random. This indicates that smaller position embeddings are trained more than larger ones (which occur less frequently). We suspect that embedding similarity correlates with the number of gradient updates a single position embedding receives. Positions 0, 1 and 128, 129 receive a gradient update in every step and can thus be considered an average of all gradient updates (up to random initialization). This is potentially one reason for the diagonal pattern in the top panel.

3.3 Corpus Comparability

So far we have trained on a parallel corpus. Now we show what happens with a merely comparable corpus. The first half of the training corpus is used for English and the other half for Fake-English. To mitigate the reduced amount of training data we train for twice as many epochs. Table 2 shows that multilinguality indeed decreases as the training cor-

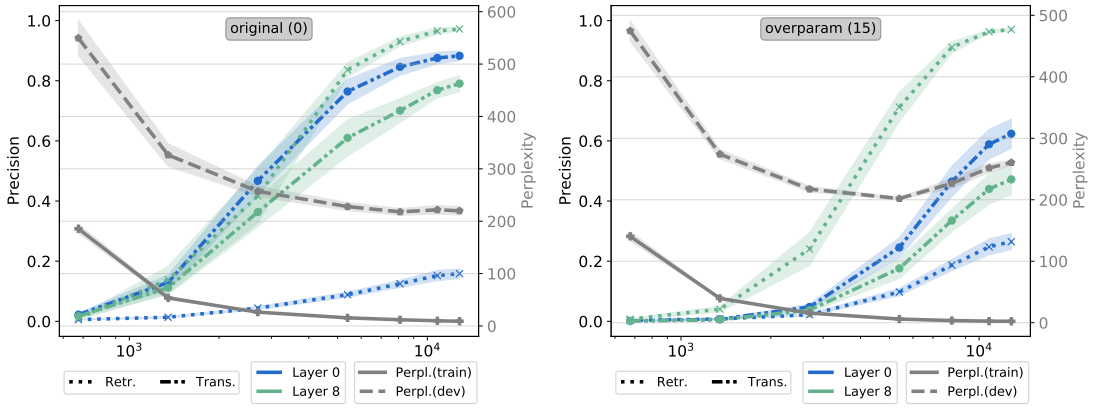


Figure 6: The longer a model is trained, the more multilingual it gets. x-axis shows training steps. Alignment F_1 is not shown as the models use shared position embeddings. Lines show mean and shaded areas show standard deviation across 5 random seed.

pus becomes non-parallel. This suggests that the more comparable a training corpus is across languages the higher the multilinguality. Note, however, that the models fit the training data worse and do not generalize as well as the original model.

3.4 Multilinguality During Training

One central hypothesis is that BERT becomes multilingual at the point at which it is forced to use its parameters efficiently. We argue that this point depends on several factors including the number of parameters, training duration, "complexity" of the data distribution and how easily common structures across language spaces can be aligned. The latter two are difficult to control for. We provided insights that two languages with identical structure but inverted word order are harder to align. Figure 6 analyzes the former two factors and shows model fit and multilinguality for the small and large model settings over training steps.

Generally, multilinguality rises very late at a stage where model fit improvements are flat. In fact, most of multilinguality in the overparameterized setting (15) arises once the model starts to overfit and perplexity on the development set goes up. The original setting (0) has far fewer parameters. We hypothesize that it is forced to use its parameters efficiently and thus multilinguality scores rise much earlier when both training and development perplexity are still going down.

Although this is a very restricted experimental setup it indicates that having multilingual models is a trade-off between good generalization and high degree of multilinguality. By overfitting a model one could achieve high multilinguality. Conneau et al. (2020a) introduced the concept of "curse of multilinguality" and found that the number of parameters should be increased with the number of languages. Our results indicate that too many parameters can also harm multilinguality. However, in practice it is difficult to create a model with so many parameters that it is overparameterized when being trained on 104 Wikipedias.

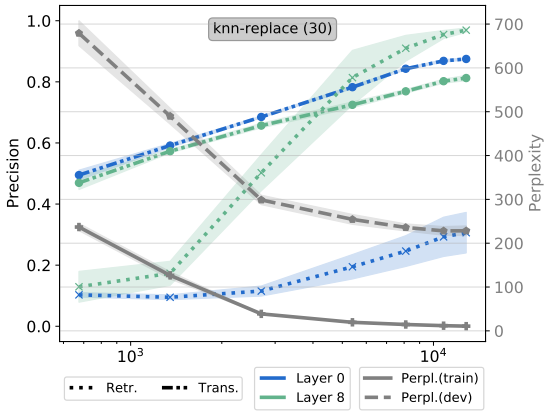


Figure 7: With knn-replace multilinguality rises earlier. Alignment F_1 is not shown as the model uses shared position embeddings.

Rönnqvist et al. (2019) found that current multilingual BERT models may be undertrained. This is consistent with our findings that multilinguality arises late in the training stage.

4 Improving Multilinguality

So far we have tried to break BERT's multilinguality. Now we turn to exploiting our insights for improving it. mBERT has shared position embeddings, shared special tokens and we cannot change linguistic properties of languages. Our results on

overparameterization suggest that smaller models become multilingual faster. However, mBERT may already be considered underparameterized given that it is trained on 104 large Wikipedias.

One insight we can leverage for the masking procedure is *no-random*: replacing masked words with random tokens. *We propose to introduce a fourth masking option:* replacing masked tokens with semantically similar words from other languages. To this end we train static fastText embeddings (Bojanowski et al., 2017) on the training set and then project them into a common space using VecMap (Artetxe et al., 2018). We use this crosslingual space to replace masked tokens with nearest neighbors from the other language. Each masked word is then replaced with the probabilities $(p_{[\text{mask}]}, p_{[\text{id}]}, p_{[\text{rand}]}, p_{[\text{knn}]}) = (0.5, 0.1, 0.1, 0.3)$, i.e., in 30% of the cases masked words get replaced with the nearest neighbor from the multilingual static embedding space. Note that this procedure (including VecMap) is fully unsupervised (i.e., no parallel data or dictionary required). We call this method *knn-replace*. Conneau et al. (2020b) performed similar experiments by creating code switched data and adding it to the training data. However, we only replace masked words.

Figure 7 shows the multilinguality score and model fit over training time. Compared to the original model in Figure 6, retrieval and translation have higher scores earlier. Towards the end multilinguality scores become similar, with knn-replace outperforming the original model (see Table 1). This finding is particularly important for training BERT on large amounts of data. Given how expensive training is, it may not be possible to train a model long enough to obtain a high degree of multilinguality. Longer training incurs the risk of overfitting as well. Thus achieving multilinguality early in the training process is valuable. Our new masking strategy has this property.

5 Real Data Experiments

5.1 XNLI

We have presented experiments on a small corpus with English and Fake-English. Now we provide results on real data. Our setup is similar to (K et al., 2020): we train a multilingual BERT model on English, German and Hindi. As training corpora we sample 1GB of data from Wikipedia (except for Hindi, as its size is <1GB) and pretrain the model for 2 epochs/140k steps with batch size 256 and learning rate 1e-4. In this section, we use BERT-base, not BERT-small because we found that BERT-small with less than 1M parameters performs poorly in a larger scale setup. The remaining model and training parameters are the same as before. Each language has its own vocabulary with size 20k. We then evaluate the pretrained models on XNLI (Conneau et al., 2018). We finetune the pretrained models on English XNLI (3 epochs, batch size 32, learning rate 2e-5, following Devlin et al. (2019)). Then the model is evaluated on English. In addition, we do a zero-shot evaluation on German and Hindi.

ID	Description	ENG	DEU	HIN
0-base	original	**.75** $_{.00}$	.57 $_{.02}$	.45 $_{.01}$
3-base	inv-order[DEU]	**.75** $_{.00}$	.41 $_{.01}$	.46 $_{.04}$
8-base	lang-pos;shift-special;no-random	.74 $_{.00}$	.37 $_{.02}$	.38 $_{.02}$
30-base	knn-replace	.74 $_{.01}$	**.61** $_{.01}$	**.54** $_{.00}$
mBERT	Results by (Hu et al., 2020)	.81	.70	.59

Table 3: Accuracy on XNLI test for different model settings. Shown is the mean and standard deviation (subscript) across three random seeds. All models have the same architecture as BERT-base, are pretrained on Wikipedia data and finetuned on English XNLI training data. mBERT was pretrained longer and on much more data and has thus higher performance. Best non-mBERT performance in bold.

Table 3 presents accuracy on XNLI test. Compared to mBERT, accuracy is significantly lower but reasonable on English (.75 vs. .81) – we pretrain on far less data. ID 0 shows high multilinguality with 0-shot accuracies .57 and .45. Inverting the order of German has little effect on HIN, but DEU drops significantly (majority baseline is .33). Our architectural modifications (8) harm both HIN and DEU. The proposed *knn-replace* model exhibits the strongest degree of multilinguality, boosting the 0-shot accuracy in DEU / HIN by 4% / 9%. Note that to accommodate noise in the real world data, we randomly replace with one of the five nearest neighbors (not the top nearest neighbor). This indicates that *knn-replace* is useful for real world data and that our prior findings transfer to larger scale settings.

6 Related Work

There is a range of prior work analyzing the reason for BERT's multilinguality. Singh et al. (2019) show that BERT stores language representations in different subspaces and investigate how subword tokenization influences multilinguality. Artetxe et al. (2020) show that neither a shared vocabulary nor

joint pretraining is essential for multilinguality. K et al. (2020) extensively study reasons for multilinguality (e.g., researching depth, number of parameters and attention heads). They conclude that depth is essential. They also investigate language properties and conclude that structural similarity across languages is important, without further defining this term. Last, Conneau et al. (2020b) find that a shared vocabulary is not required. They find that shared parameters in the top layers are required for multilinguality. Further they show that different monolingual BERT models exhibit a similar structure and thus conclude that mBERT somehow aligns those isomorphic spaces. They investigate having separate embedding look-ups per language (including position embeddings and special tokens) and a variant of avoiding cross-language replacements. Their method "extra anchors" yields a higher degree of multilinguality. In contrast to this prior work, we investigate multilinguality in a clean laboratory setting, investigate the interaction of architectural aspects and research new aspects such as overparameterization or inv-order.

Other work focuses on creating better multilingual models. Mulcaire et al. (2019) proposed a method to learn multilingual contextual representations. Conneau and Lample (2019) introduce the translation modeling objective. Conneau et al. (2020a) propose XLM-R. They introduce the term "curse of multilinguality" and show that multilingual model quality degrades with an increased number of languages given a fixed number of parameters. This can be interpreted as the minimum number of parameters required whereas we find indications that models that are too large can be harmful for multilinguality as well. Cao et al. (2020) improve the multilinguality of mBERT by introducing a regularization term in the objective, similar to the creation of static multilingual embedding spaces. Huang et al. (2019) extend mBERT pretraining with three additional tasks and show an improved overall performance. More recently, better multilinguality is achieved by Pfeiffer et al. (2020) (adapters) and Chi et al. (2020) (parallel data). We propose a simple extension to make mBERT more multilingual; it does not require additional supervision, parallel data or a more complex loss function – in contrast to this prior work.

Finally, many papers find that mBERT yields competitive zero-shot performance across a range of languages and tasks such as parsing and NER (Pires et al., 2019; Wu and Dredze, 2019), word alignment and sentence retrieval (Libovický et al., 2019) and language generation (Rönnqvist et al., 2019); Hu et al. (2020) show this for 40 languages and 9 tasks. Wu and Dredze (2020) consider the performance on up to 99 languages for NER. In contrast, Lauscher et al. (2020) show limitations of the zero-shot setting and Zhao et al. (2020) observe poor performance of mBERT in reference-free machine translation evaluation. Prior work here focuses on investigating the degree of multilinguality, not the reasons for it.

7 Conclusion

We investigated which architectural and linguistic properties are essential for BERT to yield crosslingual representations. The main takeaways are: **i)** Shared position embeddings, shared special tokens, replacing masked tokens with random tokens and a limited amount of parameters are necessary elements for multilinguality. **ii)** Word order is relevant: BERT is not multilingual with one language having an inverted word order. **iii)** The comparability of training corpora contributes to multilinguality. We show that our findings transfer to larger scale settings. We experimented with a simple modification to obtain stronger multilinguality in BERT models and demonstrate its effectiveness on XNLI. We considered a fully unsupervised setting without any crosslingual signals. In future work we plan to incorporate crosslingual signals as Vulić et al. (2019) argue that a fully unsupervised setting is hard to motivate.

Acknowledgements

We gratefully acknowledge funding through a Zentrum Digitalisierung.Bayern fellowship awarded to the first author. This work was supported by the European Research Council (# 740516). We thank Mengjie Zhao, Nina Pörner, Denis Peskov and the anonymous reviewers for fruitful discussions and valuable comments.

References

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, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

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.

Alexandra Birch and Miles Osborne. 2011. Reordering metrics for MT. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1027–1035, Portland, Oregon, USA. Association for Computational Linguistics.

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.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. 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. In *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. 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.

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

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. 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 2485–2494, Hong Kong, China. Association for Computational Linguistics.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. *arXiv preprint arXiv:2005.00633*.

Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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).

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

Phoebe Mulcaire, Jungo Kasai, and Noah A. Smith. 2019. Polyglot contextual representations improve crosslingual transfer. 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 3912–3918, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew 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.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

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, Florence, Italy. Association for Computational Linguistics.

Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. 2019. Is multilingual BERT fluent in language generation? In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 29–36, Turku, Finland. Linköping University Electronic Press.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *J. Artif. Int. Res.*, 65(1):569–630.

Masoud Jalili Sabet, Philipp Dufter, and Hinrich Schütze. 2020. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. *arXiv preprint arXiv:2004.08728*.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Jasdeep Singh, Bryan McCann, Richard Socher, and Caiming Xiong. 2019. BERT is not an interlingua and the bias of tokenization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 47–55, Hong Kong, China. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 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. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Wei Zhao, Goran Glavaš, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1656–1671, Online. Association for Computational Linguistics.

A Additional Details on Methods

A.1 Word Translation Evaluation

Word translation is evaluated in the same way as sentence retrieval. This section provides additional details.

For each token in the vocabulary $w^{(k)}$ we feed the "sentence" "[CLS] $\{ w^{(k)}\}$ [SEP]" to the BERT model to obtain the embeddings $\mathcal{E}(w^{(k)}) \in \mathbb{R}^{3\times d}$ from the l-th layer of BERT for $k \in \{\text{eng}, \text{fake}\}$. Now, we extract the word embedding by taking the second vector (the one corresponding to $w^{(k)}$) and denote it by $e_w^{(k)}$. Computing cosine similarities between English and Fake-English tokens yields the similarity matrix $R \in \mathbb{R}^{m\times m}$ where $R_{ij} = \text{cosine-sim}(e_i^{(\text{eng})}, e_j^{(\text{fake})})$ for m tokens in the vocabulary of one language (in our case 2048).

Given an English query token $s_i^{(\text{eng})}$, we obtain the retrieved tokens in Fake-English by ranking them according to similarity. Note that we can do the same with Fake-English as query language. We report the mean precision of these directions that is computed as

$$\tau = \frac{1}{2m}\sum_{i=1}^{m} \mathbb{1}_{\arg\max_l R_{il}=i} + \mathbb{1}_{\arg\max_l R_{li}=i}.$$

A.2 inv-order

Assume the sentence "He ate wild honey ." exists in the corpus. The tokenized version is [He, ate, wild, hon, ##e, ##y, .] and the corresponding Fake-English sentence is [::He, ::ate, ::wild, ::hon, ::##e, ::##y, ::.]. If we apply the modification inv-order we always invert the order of the Fake-English sentences, thus the model only receives the sentence [::., ::##y, ::##e, ::hon, ::wild, ::ate, ::He].

A.3 knn-replace

We use the training data to train static word embeddings for each language using the tool fastText. Subsequently we use VecMap (Artetxe et al., 2018) to map the embedding spaces from each language into the English embedding space, thus creating a multilingual static embedding space. We use VecMap without any supervision.

During MLM-pretraining of our BERT model 15% of the tokens are randomly selected and "masked". They then get either replaced by "[MASK]" (50% of the cases), remain the same (10% of the cases), get replaced by a random other token (10% of the cases) or we replace the token with one of the five nearest neighbors (in the fake-English setup only with the nearest neighbor) from another language (30% of the cases). Among those five nearest neighbors we pick one randomly. In case more than one other language is available we pick one randomly.

Lang.	Kendall's Tau Distance	XNLI Acc.
en	1.0	81.4
ar	0.72	64.9
de	0.74	71.1
fr	0.80	73.8
ru	0.72	69.0
th	0.71	55.8
ur	0.59	58.0
zh	0.68	69.3
bg	0.75	68.9
el	0.77	66.4
es	0.76	74.3
hi	0.58	60.0
sw	0.73	50.4
tr	0.47	61.6
vi	0.78	69.5

Table 4: Kendall's Tau word order metric and XNLI zero-shot accuracies.

B Additional Non-central Results

B.1 Model 17

One might argue that our model 17 in Table 1 of the main paper is simply not trained enough and thus not multilingual. However, Table 10 shows that even when continuing to train this model for a long time no multilinguality arises. Thus in this configuration the model has enough capacity to model the languages independently of each other – and due to the modifications apparently no incentive to try to align the language representations.

B.2 Word Order in XNLI

To verify whether similar word order across languages influences the multilinguality we propose to compute a word reordering metric and correlate this metric with the performance of 0-shot transfer capabilities of mBERT. To this end we consider the performance of mBERT on XNLI. We follow Birch and Osborne (2011) in computing word reordering metrics between parallel sentences (XNLI is a parallel corpus). More specifically we compute the Kendall's tau metric. To this end, we compute word alignments between two sentences using the Match algorithm by Sabet et al. (2020), which directly yield a permutation between sentences as

Scenario	Runtime
pretrain small BERT model on Easy-to-Read-Bible, 100 epochs	$\sim 35m$
pretrain large BERT model (BERT-base) on Easy-to-Read-Bible, 100 epochs	$\sim 4h$
pretrain large BERT model (BERT-base) on Wikipedia sample, 1 epoch	$\sim 2.5days$

Table 5: Runtime on a single GPU.

Model	Parameters
Standard Configuration ("Small model")	$1M$
BERT-Base / Overparameterized Model / "Large model"	$88M$
Real data model (BERT-Base with larger vocabulary)	$131M$
mBERT	$178M$

Table 6: Number of parameters for our used models.

required by the distance metric. We compute the metric on 2500 sentences from the development data of XNLI and average it across sentences to get a single score per language. The scores and XNLI accuracies are in Table 4.

The Pearson correlation between Kendall's tau metric and the XNLI classification accuracy in a zero-shot scenario (mBERT only finetuned on English and tested on all other languages) is 46% when disregarding English and 64% when including English. Thus there is a some correlation observable. This indicates that zero-shot performance of mBERT might also rely on similar word order across languages. We plan to extend this experiment to more zero-shot results and examine this effect more closely in future work.

B.3 Larger Position Similarity Plots

We provide larger versions of our position similarity plots in Figure 8.

C Reproducibility Information

C.1 Data

Table 7 provides download links to data.

C.2 Technical Details

The number of parameters for each model are in Table 6.

We did all computations on a server with up to 40 Intel(R) Xeon(R) CPU E5-2630 v4 CPUs and 8 GeForce GTX 1080Ti GPU with 11GB memory. No multi-GPU training was performed. Typical runtimes are reported in Table 5.

Used third party systems are shown in Table 8.

C.3 Hyperparameters

We show an overview on hyperparameters in Table 9. If not shown we fall back to default values in the systems.

Name	Languages	Description	Size	Link
XNLI (Conneau et al., 2018)	English, German, Hindi	Natural Language Inference Dataset. We use the English training set and English, German and Hindi test set.	392703 sentence pairs in train, 5000 in test, 2500 in dev per language.	https://cims.nyu.edu/~sbowman/xnli/
Wikipedia	English, German, Hindi	We use 1GB of randomly sampled data from a Wikipedia dump downloaded in October 2019.	8.5M sentences for ENG, 9.3M for DEU and 800K for HIN.	download.wikimedia.org/[X]wiki/latest/[X]wiki-latest-pages-articles.xml.bz2
Bible (Mayer and Cysouw, 2014)	English	We use the editions Easy-to-Read and King-James-Version.	We use all 17178 sentences in Easy-to-Read (New Testament) and the first 10000 sentences of King-James in the Old Testament.	n/a

Table 7: Overview on datasets.

System	Parameter	Value
Vecmap	Code URL	https://github.com/artetxem/vecmap.git
	Git Commit Hash	b82246f6c249633039f67fa6156e51d852bd73a3
	Version	0.9.1
fastText	Code URL	https://github.com/facebookresearch/fastText/archive/v0.9.1.zip
	Embedding Dimension	300
Transformers	Version	2.8.0
Tokenizers	Version	0.5.2
NLTK	Version	3.4.5

Table 8: Overview on third party systems used.

Parameter	Value
Hidden size	64; 768 for large models (i.e., overparameterized and those used for XNLI) derived from BERT-based configuration
Intermediate layer size	256; 3072 for large models
Number of attention heads	1; 12 for large models
Learning rate	$2e-3$ (chosen out of $1e-4$, $2e-4$, $1e-3$, $2e-r$, $1e-2$, $2e-2$ via grid search; criterion: perplexity); $1e-4$ for large models, same as used in (Devlin et al., 2019)
Weight decay	0.01 following (Devlin et al., 2019)
Adam epsilon	$1e-6$ following (Devlin et al., 2019)
Random Seeds	0, 42, 43, 100, 101; For single runs: 42. For real data experiments: 1,42 and 100.
Maximum input length after tokenization	128
Number of epochs	100 unless indicated otherwise. (chosen out of 10, 20, 50, 100, 200 via grid search; criterion: perplexity)
Number of warmup steps	50
Vocabulary size	4096; 20000 per language for the XNLI models
Batch size	256 for pretraining (for BERT-Base models 16 with 16 gradient accumulation steps), 32 for finetuning

Table 9: Model and training parameters during pretraining.

ID	Description	Num. Epochs	Mult.-score μ	Layer 0 Align. F_1	Layer 0 Retr. ρ	Layer 0 Trans. τ	Layer 8 Align. F_1	Layer 8 Retr. ρ	Layer 8 Trans. τ	MLM-Perpl. train	MLM-Perpl. dev
0	original	100	.70	1.00 $_{.00}$	.16 $_{.02}$	.88 $_{.02}$	1.00 $_{.00}$	.97 $_{.01}$	.79 $_{.03}$	9 $_{00.22}$	217 $_{07.8}$
17	lang-pos;shift-special;no-random;overparam	100	.00	.05 $_{.02}$	.00 $_{.00}$	.00 $_{.00}$	.05 $_{.04}$	.00 $_{.00}$	.00 $_{.00}$	2 $_{00.02}$	270 $_{20.1}$
17	lang-pos;shift-special;no-random;overparam	250	.00	.06 $_{.02}$	.00 $_{.00}$	.00 $_{.00}$	.06 $_{.05}$	.00 $_{.00}$	.00 $_{.00}$	1 $_{00.00}$	1111 $_{30.7}$

Table 10: Even when continuing the training for a long time overparameterized models with architectural modifications do not become multilingual.

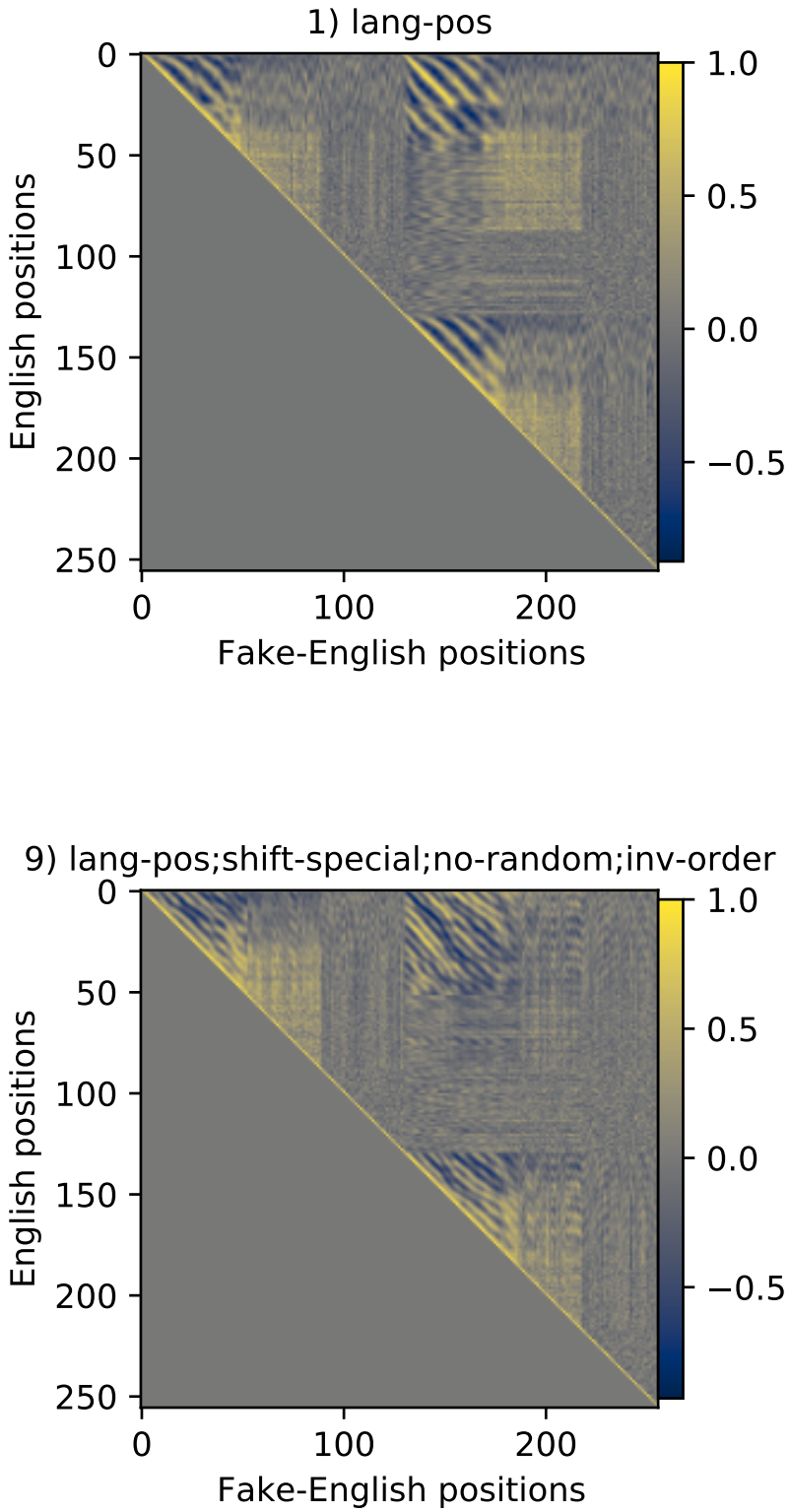


Figure 8: Cosine similarity of position embeddings. IDs 0-127 are used for English, 128-255 for Fake-English. .

Chapter 7

Multilingual LAMA: Investigating Knowledge in Multilingual Pretrained Language Models

Multilingual LAMA: Investigating Knowledge in Multilingual Pretrained Language Models

Nora Kassner*, **Philipp Dufter***, **Hinrich Schütze**
Center for Information and Language Processing (CIS), LMU Munich, Germany
{kassner,philipp}@cis.lmu.de

Abstract

Recently, it has been found that monolingual English language models can be used as knowledge bases. Instead of structural knowledge base queries, masked sentences such as "Paris is the capital of [MASK]" are used as probes. We translate the established benchmarks TREx and GoogleRE into 53 languages. Working with mBERT, we investigate three questions. (i) Can mBERT be used as a multilingual knowledge base? Most prior work only considers English. Extending research to multiple languages is important for diversity and accessibility. (ii) Is mBERT's performance as knowledge base language-independent or does it vary from language to language? (iii) A multilingual model is trained on more text, e.g., mBERT is trained on 104 Wikipedias. Can mBERT leverage this for better performance? We find that using mBERT as a knowledge base yields varying performance across languages and pooling predictions across languages improves performance. Conversely, mBERT exhibits a language bias; e.g., when queried in Italian, it tends to predict Italy as the country of origin.

1 Introduction

Pretrained language models (LMs) (Peters et al., 2018; Howard and Ruder, 2018; Devlin et al., 2019) can be finetuned to a variety of natural language processing (NLP) tasks and generally yield high performance. Increasingly, these models and their generative variants are used to solve tasks by simple text generation, without any finetuning (Brown et al., 2020). This motivated research on how much knowledge is contained in LMs: Petroni et al. (2019) used models pretrained with masked language to answer fill-in-the-blank templates such as "Paris is the capital of [MASK]."

	Query	Two most frequent predictions
en	X was created in MASK.	[Japan (170), Italy (56), . . .]
de	X wurde in MASK erstellt.	[Deutschland (217), Japan (70), . . .]
it	X è stato creato in MASK.	[Italia (167), Giappone (92), . . .]
nl	X is gemaakt in MASK.	[Nederland (172), Italië (50), . . .]
en	X has the position of MASK.	[bishop (468), God (68), ...]
de	X hat die Position MASK.	[WW (261), Ratsherr (108), ...]
it	X ha la posizione di MASK.	[pastore (289), papa (138), ...]
nl	X heeft de positie van MASK.	[burgemeester (400), bisschop (276) , ...]

Table 1: Language bias when querying (TyQ) mBERT. Top: For an Italian cloze question, Italy is favored as country of origin. Bottom: There is no overlap between the top-ranked predictions, demonstrating the influence of language – even though the facts are the same: the same set of triples is evaluated across languages. Table 3 shows that pooling predictions across languages addresses bias and improves performance. WW = "Wirtschaftswissenschaftler".

This research so far has been exclusively on English. In this paper, we focus on using *multilingual* pretrained LMs as knowledge bases. Working with mBERT, we investigate three questions. (i) Can mBERT be used as a multilingual knowledge base? Most prior work only considers English. Extending research to multiple languages is important for diversity and accessibility. (ii) Is mBERT's performance as knowledge base language-independent or does it vary from language to language? To answer these questions, we translate English datasets and analyze mBERT for 53 languages. (iii) A multilingual model is trained on more text, e.g., BERT's training data contains the English Wikipedia, but mBERT is trained on 104 Wikipedias. Can mBERT leverage this fact? Indeed, we show that pooling across languages helps performance.

In summary our contributions are: **i)** We automatically create a multilingual version of TREx and GoogleRE covering 53 languages. **ii)** We use an alternative to fill-in-the-blank querying – ranking entities of the type required by the template (e.g., cities) – and show that it is a better tool

* Equal contribution - random order.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 3250–3258
April 19 - 23, 2021.

to investigate knowledge captured by pretrained LMs. **iii)** We show that mBERT answers queries across languages with varying performance: it works reasonably for 21 and worse for 32 languages. **iv)** We give evidence that the query language affects results: a query formulated in Italian is more likely to produce Italian entities (see Table 1). **v)** Pooling predictions across languages improves performance by large margins and even outperforms monolingual English BERT. Code and data are available online (https://github.com/norakassner/mlama).

2 Data

2.1 LAMA

We follow the LAMA setup introduced by Petroni et al. (2019). More specifically, we use data from TREx (Elsahar et al., 2018) and GoogleRE.[1] Both consist of triples of the form (object, relation, subject). The underlying idea of LAMA is to query knowledge from pretrained LMs using templates without any finetuning: the triple (Paris, capital-of, France) is queried with the template "Paris is the capital of [MASK]." In LAMA, TREx has 34,039 triples across 41 relations, GoogleRE 5528 triples and 3 relations. Templates for each relation have been manually created by Petroni et al. (2019). We call all triples from TREx and GoogleRE together *LAMA*.

LAMA has been found to contain many "easy-to-guess" triples; e.g., it is easy to guess that a person with an Italian sounding name is born in Italy. *LAMA-UHN* is a subset of triples that are hard to guess introduced by Poerner et al. (2020).

2.2 Translation

We translate both entities and templates. We use Google Translate to translate templates in the form "[X] is the capital of [Y]". After translation, all templates were checked for validity (i.e., whether they contain "[X]", "[Y]" exactly once) and corrected if necessary. In addition, German, Hindi and Japanese templates were checked by native speakers to assess translation quality (see Table 2). To translate the entity names, we used Wikidata and Google knowledge graphs.

mBERT covers 104 languages. Google Translate covers 77 of these. Wikidata and Google Knowledge Graph do not provide entity translations for all languages and not all entities are contained in the knowledge graphs. For English we can find a total of 37,498 triples which we use from now on. On average, 34% of triples could be translated (macro average over languages). We only consider languages with a coverage above 20%, resulting in the final number of languages we include in our study: 53. The macro average of translated triples in these 53 languages is 43%. Figure 1 gives statistics. We call the translated dataset *mLAMA*.

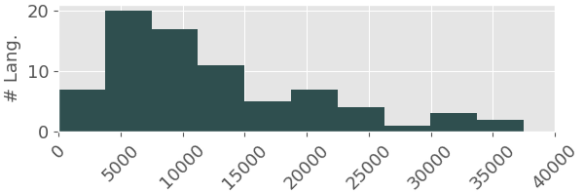


Figure 1: x-axis is the number of translated triples, y-axis the number of languages. There are 39,567 triples in the original LAMA (TREx and GoogleRE).

3 Experiments

3.1 Model

We work with mBERT (Devlin et al., 2019), a model pretrained on the 104 largest Wikipedias. We denote mBERT queried in language x as mBERT[x]. As comparison we use the English BERT-Base model and refer to it as BERT. In initial experiments with XLM-R (Conneau et al., 2020) we observed worse performance, similar to Jiang et al. (2020a). Thus, for simplicity we only report results on mBERT.

3.2 Typed and Untyped Querying

Petroni et al. (2019) use templates like "Paris is the capital of [MASK]" and give $\arg\max_{w \in \mathcal{V}} p(w|t)$ as answer where V is the vocabulary of the LM and $p(w|t)$ is the (log-)probability that word w gets predicted in the template t. Thus the object of a triple must be contained in the vocabulary of the language model. This has two drawbacks: it reduces the number of triples that can be considered drastically and hinders performance comparisons across LMs with different vocabularies. We refer to this procedure as *UnTyQ*.

We propose to use typed querying, *TyQ*: for each relation a candidate set $\mathcal{C}$ is created and the prediction becomes $\arg\max_{c \in \mathcal{C}} p(c|t)$. For templates like "[X] was born in [MASK]", we know which entity type to expect, in this case cities. We observed that (English-only) BERT-base predicts city

[1] code.google.com/archive/p/relation-extraction-corpus/

names for MASK whereas mBERT predicts years for the same template. TyQ prevents this.

We choose as $\mathcal{C}$ the set of objects across all triples for a single relation. The candidate set could also be obtained from an entity typing system (e.g., (Yaghoobzadeh and Schütze, 2016)), but this is beyond the scope of this paper. Variants of TyQ have been used before (Xiong et al., 2020).

3.3 Singletoken vs. Multitoken Objects

Assuming that objects are in the vocabulary (Petroni et al., 2019) is a restrictive assumption, even more in the multilingual case as e.g., "Hamburg" is in the mBERT vocabulary, but French "Hambourg" is tokenized to ["Ham", "##bourg"]. We consider multitoken objects by including multiple [MASK] tokens in the templates. For both TyQ and UnTyQ we compute the score that a multitoken object is predicted by taking the average of the log probabilities for its individual tokens.

Given a template t (e.g., "[X] was born in [Y].") let t_1 be the template with one mask token, (i.e., "[X] was born in [MASK].") and t_k be the template with k mask tokens (i.e., "[X] was born in [MASK] [MASK] ... [MASK]."). We denote the log probability that the token $w \in V$ is predicted at ith mask token as $p(m_i = w|t_k)$, where V is the vocabulary of the LM. To compute $p(e|t)$ for an entity e that is tokenized into l tokens $\epsilon_1, \epsilon_2, \ldots, \epsilon_l$ we simply average the log probabilities across tokens:

$$p(e|t) = \frac{1}{l} \sum_{i=1}^{l} p(m_i = \epsilon_i|t_l).$$

If k is the maximum number of tokens of any entity $e \in \mathcal{C}$ gets split into, we consider all templates $t_1, \ldots, t_k$, with C being the candidate set. The prediction is then the word with the highest average log probability across all templates $t_1, \ldots, t_k$.

Note that for UnTyQ the space of possible predictions is $V \times V \times \cdots \times V$ whereas for TyQ it is the candidate set $\mathcal{C}$.

3.4 Evaluation

We compute precision at one for each relation, i.e., $1/|T| \sum_{t \in T} \mathbb{1}\{\hat{t}_{object} = t_{object}\}$ where T is the set of all triples and $\hat{t}_{object}$ is the object predicted by TyQ or UnTyQ. Note that T is different for each language. Our final measure (p1) is then the precision at one averaged over relations (i.e., macro average). Results for multiple languages are the macro average p1 across languages.

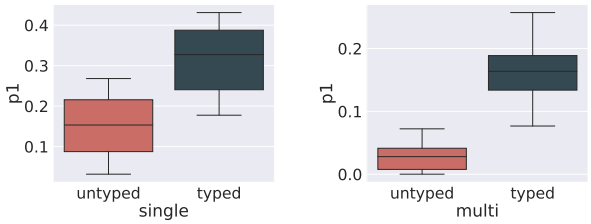


Figure 2: Distribution of p1 scores for 53 languages in UnTyQ vs. TyQ. Left: singletoken (object = 1 token). Right: multitoken (object > 1 token).

4 Results and Discussion

We first investigate TyQ and UnTyQ and find that TyQ is better suited for investigating knowledge in LMs. After exploring the translation quality, we use TyQ on mLAMA and observe rather stable performance for 21 and poor performance for 32 languages. When investigating the languages more closely, we find that prediction results highly depend on the language. Finally, we validate our initial hypothesis that mBERT can leverage its multilinguality by pooling predictions: pooling indeed performs better.

4.1 UnTyQ vs. TyQ

Figure 2 shows the distribution of p1 scores for single and multitoken objects. As expected, TyQ works better, both for single and multitoken objects. With UnTyQ, performance not only depends on the model's knowledge, but on at least three extraneous factors: **(i)** Does the model understand the type constraints of the template (e.g., in "X is the capital of Y", Y must be a country)? **(ii)** How "fluent" a substitution is an object under linguistic constraints (e.g., morphology) that can be viewed as orthogonal to knowledge? Many English templates cannot be translated into a single template in many languages, e.g., "in X" (with X a country) has different translations in French: "à Chypre", "au Mexique", "en Inde". But the LAMA setup requires a single template. By enforcing the type, we reduce the number of errors that are due to surface fluency. **(iii)** The inadequacy of the original LAMA setup for multitoken answers. Figure 2 (right) shows that the original UnTyQ struggles with multitokens (mean p1 .03 vs. .17 for TyQ).

Overall, TyQ allows us to focus the evaluation on the core question: what knowledge is contained in LMs? From now on, we report numbers in the TyQ setting.

Manual template tuning or automatic template

	machine translated	manually corrected		manually paraphrased	
de	18.1	19.4	(6)	20.9	(18)
hi	5.4	6.2	(14)	6.2	(1)
ja	0.4	0.4	(14)	0.7	(5)

Table 2: Effect of manual template modification on UnTyQ. Shown is p1, number of templates modified (in brackets). Templates are modified to correct mistakes from machine translation and paraphrased to achieve the correct object type. Manual template correction has a small effect on UnTyQ.

mining (Jiang et al., 2020b) has been investigated in the literature to approach the typing problem. We had native speakers check templates for German, Hindi and Japanese, correct mistakes in the automatic translation and paraphrase the template to obtain predictions with the correct type. Table 2 shows that corrections do not yield strong improvements. We conclude that template modifications are not an effective solution for the typing problem.

4.2 Translation Quality

Contemporaneous work by Jiang et al. (2020a) provides manual translations of LAMA templates for 23 languages respecting grammatical gender and inflection constraints. We evaluate our machine translated templates by comparing performance on a common subset of 14 languages using TyQ querying on the TREx subset. Surprisingly, we find a performance difference of 1 percentage points (0.23 vs. 0.24, p1 averaged over languages) in favor of the machine translated templates. This indicates that the machine translated templates in combination with TyQ exhibit comparable performance but come with the benefit of larger language coverage (53 vs. 23 languages).

4.3 Multilingual Performance

In mLAMA, not all triples are available in all languages. Thus absolute numbers are not comparable across languages and we adopt a relative performance comparison: we report p1 of a model-language combination divided by p1 of mBERT's performance in English (mBERT[en]) on the exact same set of triples and call this *rel-p1*. A rel-p1 score of 0.5 for mBERT[fi] means that p1 of mBERT on Finnish is half of mBERT[en]'s performance on the same triples. rel-p1 of English BERT is usually greater than 1 as monolingual BERT tends to outperform mBERT[en].

Figure 3 shows that mBERT performs reasonably well for 21 languages, but for 32 languages

	LAMA	LAMA-UHN
BERT	38.5	29.0
mBERT[en]	35.0	25.7
mBERT[pooled]	**41.1**	**32.1**

Table 3: p1 for BERT, mBERT queried in English, mBERT pooled on LAMA and LAMA-UHN.

rel-p1 is less than 0.6 (i.e., their p1 is 60% of English's p1). We conclude that mBERT does not exhibit a stable performance across languages. The variable performance (from 20% to almost 100% rel-p1) indicates that mBERT has no common representation for, say, "Paris" across languages, i.e., mBERT representations are language-dependent.

4.4 Bias

If mBERT captured knowledge independent of language, we should get similar answers across languages for the same relation. However, Table 1 shows that mBERT exhibits language-specific biases; e.g., when queried in Italian, it tends to predict Italy as the country of origin. This effect occurs for several relations: Table 4 in the supplementary presents data for ten relations and four languages.

4.5 Pooling

We investigate pooling of predictions across languages by picking the object predicted by the majority of languages. Table 3 shows that pooled mBERT outperforms mBERT[en] by 6 percentage points on LAMA, presumably in part because the language-specific bias is eliminated. mBERT[pooled] even outperforms BERT by 3 percentage points on LAMA-UHN. This indicates that mBERT can leverage the fact that it is trained on 104 Wikipedias vs. just one and even outperforms the much stronger model BERT.

5 Related Work

Petroni et al. (2019) first asked the question: can pretrained LMs function as knowledge bases? Subsequent analyses focused on different aspects, such as negation (Kassner and Schütze, 2020), easy to guess names (Poerner et al., 2020), integrating adapters (Wang et al., 2020) or finding alternatives to a "fill-in-the-blank" approach with single-token answers (Bouraoui et al., 2020; Heinzerling and Inui, 2020; Jiang et al., 2020b). Other work combines pretrained LM with information retrieval (Guu et al., 2020; Lewis et al., 2020a; Izacard and Grave, 2020; Kassner and Schütze, 2020; Petroni

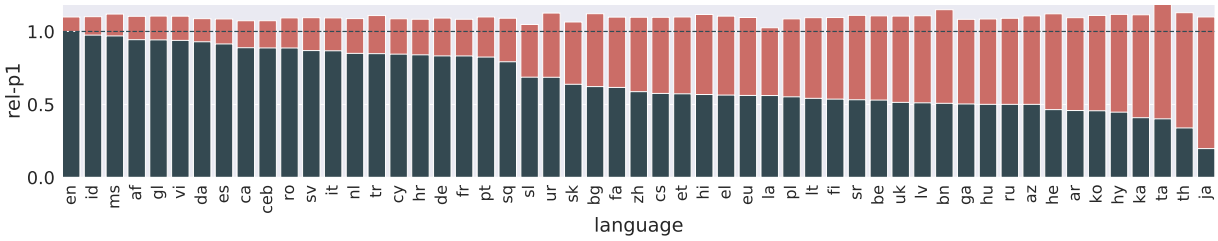


Figure 3: p1 of BERT (red) vs mBERT[x] (blue) divided by p1 of mBERT[en] on the same set of triples in each language x. mBERT captures less factual knowledge than monolingual English BERT. While performance is reasonable for 21 languages, it is below 60% for 32 languages. Dashed line is rel-p1 of mBERT[en] (by definition equal to 1.0). Performance of BERT varies slightly as the set of triples is different for each language. Note that the Wikipedia of Cebuano (ceb) consists mostly of machine translated articles.

et al., 2020). None of this work addresses languages other than English.

Multilingual models like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) perform well for zero-shot crosslingual transfer (Hu et al., 2020). However, we are not aware of any prior work that analyzed to what degree pretrained multilingual models can be used as knowledge bases. There are many multilingual question answering datasets such as XQuAD (Artetxe et al., 2020), TiDy (Clark et al., 2020), MKQA (Longpre et al., 2020) and MLQA (Lewis et al., 2020b). Usually, multilingual models are finetuned to solve such tasks. Our goal is not to improve question answering or create an alternative multilingual question answering dataset, but instead to investigate which knowledge is contained in pretrained multilingual LMs without any kind of supervised finetuning.

There is a range of alternative multilingual knowledge bases that could be used for evaluation. Those include ConceptNet (Speer et al., 2017) or BabelNet (Navigli and Ponzetto, 2010). We decided to provide a translated versions of TREx and GoogleRE for the sake of comparability across languages. By translating manually created templates and entities we can ensure comparability across languages. This is not possible for crowd-sourced databases like ConceptNet.

In contemporaneous work, Jiang et al. (2020a) create and investigate a multilingual version of LAMA. They provide human template translations for 23 languages, propose several methods for multitoken decoding and code-switching, and experiment with a number of PLMs. In contrast to their work, we investigate typed querying, focus on comparabiliy and pooling across languages, and explore language biases.

6 Conclusion

We presented mLAMA, a dataset to investigate knowledge in language models (LMs) in a multilingual setting covering 53 languages. While our results suggest that correct entities can be retrieved for many languages, there is a clear performance gap between English and, e.g., Japanese and Thai. This suggests that mBERT is not storing entity knowledge in a language-independent way. Experiments investigating language bias confirm this finding. We hope that this paper and the dataset we publish will stimulate research on investigating knowledge in LMs *multilingually* rather than just in English.

Acknowledgements

This work was supported by the European Research Council (# 740516) and the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibility for its content. The second author was supported by the Bavarian research institute for digital transformation (bidt) through their fellowship program. We thank Yannick Couzinié and Karan Tiwana for correcting the Japanese and Hindi templates. We thank the anonymous reviewers for valuable comments.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from BERT. In *The Thirty-Fourth AAAI Conference*

on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020, pages 7456–7463. AAAI Press.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jonathan H. Clark, Jennimaria Palomaki, Vitaly Nikolaev, Eunsol Choi, Dan Garrette, Michael Collins, and Tom Kwiatkowski. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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.

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.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *Computing Research Repository*, arXiv:2002.08909.

Benjamin Heinzerling and Kentaro Inui. 2020. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. *Computing Research Repository*, arXiv:2008.09036.

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.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Gautier Izacard and E. Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *ArXiv*, abs/2007.01282.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020a. X-FACTR: Multilingual factual knowledge retrieval from pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5943–5959, Online. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020b. How can we know what language models know. *Transactions of the Association for Computational Linguistics*, 8:423–438.

Nora Kassner and Hinrich Schütze. 2020. BERT-kNN: Adding a kNN search component to pretrained language models for better QA. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 3424–3430. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020b. MLQA: Evaluating cross-lingual extractive question answering. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7315–7330, Online. Association for Computational Linguistics.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2020. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *CoRR*, abs/2007.15207.

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 216–225. The Association for Computer Linguistics.

Matthew 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.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *Computing Research Repository*, arXiv:2002.01808.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2016. Intrinsic subspace evaluation of word embedding representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 236–246, Berlin, Germany. Association for Computational Linguistics.

A Language Bias

Table 4 shows the language bias for 10 relations. For each relation we aggregated the predictions across all triples and show the most common two predicted entities together with its count (in brackets). The querying language clearly affects results. The effect is drastic for relations that ask for a country (e.g., P495 or P1001). P39 yields very different results without exhibiting a clear pattern. Other relations such as P463 or P178 are rather stable.

B Data Samples

Table 4 and Table 5 show randomly sampled entries from the data.

C Pretraining Data

We investigate whether performance across languages is correlated with the amount of pretraining data for each language. To this end we investigate the number of articles per language as of January 2021 [2] and p1 for TyQ in Figure 6. We do not have access to the original pretraining data of mBERT. Thus, the number of articles we consider in the analysis might be different to the actual data used to train mBERT.

	Microsoft Word word ontwikkel deur Microsoft.
af	Aix-en-Provence is die hoofstad van Provence.
	Die hoofstad van Alpes-Maritimes is Nice.
	مات جون ديورانت بريفال في باريس.
ar	يتم إنتاج كيندل فاير بواسطة أمازون.
	اللغة الرسمية لـ كوسوفو هي الصربية.
	Madrid və Sofiya əkiz şəhərlərdir.
az	İbn ən-Nədim islam dinə bağlıdır.
	Baş qərargah Belavia Minsk -dədir.
	Афіцыйнай мовай Фінляндская праваслаўная царква з'яўляецца фінская мова.
be	Сталіцай Правінцыя Цзянсу з'яўляецца Нанкін.
	Артур Гардэн нарадзіўся ў Манчэстэр.
	Ник Филип е роден в Лондон.
bg	Наситена мазнина се състои от въглерод.
	Официалният език на Организация на обединените нации е руски език.
	মন্টানা আইডাহো এর সাথে সীমানা ভাগ করে দেয়।
bn	মৌরিতানিয়া মালি এর সাথে সীমানা ভাগ করে দেয়।
	খার্তুম এবং কায়রো হ'ল দুটি শহর।
	Rich Gannon juga en la posició quarterback.
ca	La llengua oficial de Nastola és finès.
	Nova York i Jerusalem són ciutats bessones.
	Ang Mario Aldo Montano usa ka lungsuranon sa Italya.
ceb	Ang opisyal nga sinultian sa Belhika mao ang Inolandes.
	Ang Praga ug Berlin mga kaluha nga lungsod.
	Maháródža je právní termín v Indie.
cs	Felix Magath hraje v poloze záložník.
	Philipp Eduard Anton von Lenard pracuje v oblasti fyzika.
	Mae Maes Awyr Rhyngwladol Des Moines wedi'i leoli yn Iowa.
cy	Iaith swyddogol Aragón yw Sbaeneg.
	Enwir Flins-sur-Seine ar ôl Afon Seine.
	Dylan Taite blev født i Liverpool.
da	Johann Andreas Schmeller døde i München.
	Hovedstaden i Yolo County er Davis.
	The Bill wurde in Englisch geschrieben.
de	Buenaventura Sitjar starb in Kalifornien.
	Mai Jones funktioniert für British Broadcasting Corporation.
	Ρεϊνάλντο Αν παίζει μουσική όπερα.
el	Το Μπόονα Σαράγεβο βρίσκεται στο Σαράγιεβο.
	Το Μεταδιθειωδες καλιο αποτελείται από θείο.
	Vienna bread is a subclass of bread .
en	Stevie Wonder is represented by music label Motown .
	Kinji Fukasaku is Japan citizen .
	Luxemburgo comparte frontera con Francia.
es	Charles Schreiber nació en Colchester.
	Polonia comparte frontera con Alemania.
	Kohtumõistjate raamat on osa Piibel -st.
et	Iisrael hoiab diplomaatilisi suhteid Jordaania -ga.
	Sambia jagab piiri Angola -ga.
	Ameriketako Estatu Batuak -ek Albania -ekin harreman diplomatikoak mantentzen ditu.
eu	Tajikistango presidente legezko terminoa da Tajikistan -n.
	Carla Bruni Paris -n lan egiteko erabiltzen zen.
	استان آنتورپ به آنتورپ نامگذاری شده است.
fa	سوزوکی کیزاشی توسط سوزوکی تولید می شود.
	رئیسجمهور زیر کلاس سیاستمدار است.
	Pääoman Genovan maakunta pääoma on Genova.
fi	Edward Joseph Kelly syntyi Chicago.
	The Home Depot perustettiin Atlanta.
	Mel Charles est pays de Galles citoyen.
fr	Audi A5 est produit par Audi.
	Honda XR est produit par Honda.
	Fuair Walter Gay bás i Páras.
ga	Tá Ollscoil Concordia suite i Montréal.
	Is cúpla cathair iad Vín agus An Bhratasláiv.
	Hannover e Bristol son cidades xemelgas.
gl	O idioma oficial de Mercia é Lingua latina.
	Afloramento de algas está composto por Alga.
	אוניברסיטת וושינגטון נמצא ב- סיאטל.
he	ויקיפדיה הצ'כית נכתב ב- צ'כית.
	תיירות מין היא תת-סוג של תיירות.
	कीनिया युगाण्डा के साथ राजनयिक संबंध बनाए रखता है।
hi	कावेरी नदी एशिया में स्थित है।
	फ़िलीपीन्स का ध्वज फ़िलीपीन्स में एक कानूनी शब्द है।
	Istanbul i Bukurešt su gradovi blizanci.
hr	Gro Harlem Brundtland se nekada radila u Oslo.
	Izvorni jezik Ćelava pjevačica je francuski jezik.
	Gilles Grimandi Gap -ben született.
hu	Edo-kor elnevezése Edo.
	Joseph-Marie, comte Portalis anyanyelve francia.
	Իրան գտնվում է Ասիա –ում:
hy	Բարսելոնա և Դուբլին զույգ քաղաքներ են:
	Ադրբեջան -ի մայրաքաղաքն է Բաքու:

Figure 4: Three randomly sampled data entries from mLAMA per language. Due to the automatic generation of the dataset not all of them are fully correct.

[2] https://meta.wikimedia.org/wiki/List_of_Wikipedias

	en	de	nl	it
P495: “[X] was created in [Y]”	Japan (170), Italy (56)	Deutschland (217), Japan (70)	Nederland (172), Italië (50)	Italia (167), Giappone (92)
P101: “[X] works in the field of [Y]”	art (205), science (135)	Kunst (384), Film (64)	psychologie (263), kunst (120)	fisiologia (168), caccia (135)
P106: “[X] is [Y] by profession”	politician (423), composer (80)	Politiker (323), Journalist (128)	politicus (339), acteur (247)	giornalista (420), giurista (257)
P1001: “[X] is a legal term in [Y]”	India (12), Germany (11)	Deutschland (36), Russland (9)	Nederland (22), België (12)	Italia (31), Germania (16)
P39: “[X] has the position of [Y]”	bishop (468), God (68)	WW (261), Ratsherr (108)	burgemeester (400), bisschop (276)	pastore (289), papa (138)
P527 “[X] consists of [Y]”	sodium (125), carbon (88)	Wasserstof (398), C (49)	vet (216), aluminium (130)	calcio (165), atomo (96)
P1303 “[X] plays [Y]”	guitar (431), piano (165)	Gitarre (312), Klavier (204)	piano (581), harp (42)	arpa (188), pianoforte (139)
P178 “[X] is developed by [Y]”	Microsoft (177), IBM (55)	Microsoft (153), Apple (99)	Microsoft (200), Nintendo (69)	Microsoft (217), Apple (49)
P264 “[X] is represented by music label [Y]”	EMI (267), Swan (32)	EMI (202), Paramount Records (59)	EMI (225), Swan (50)	EMI (217), Swan (99)
P463 “[X] is a member of [Y]”	FIFA (126), NATO (33)	FIFA (118), NATO (38)	FIFA (157), WWE (16)	FIFA (121), NATO (36)

Table 4: Most frequent object predictions (TyQ) in different languages. Some relations exhibit language specific biases. WW = “Wirtschaftswissenschaftler”.

	Landskrona BoIS terletak di Swedia.
id	Roy Hargrove memainkan Trompet.
	Bahasa resmi Ossetia Selatan adalah Rusia.
	Federazione calcistica di Vanuatu è un membro di FIFA.
it	Mikhail Gromov funziona nel campo di geometria.
	letteratura fantasy è una sottoclasse di fantasy.
	ボルドーとカサブランカは双子の都市です。
ja	アーロン・マレーはクォーターバック位置で再生されます。
	エンツォ・フェラーリはイタリア市民です。
	ვოტკინი F შედგება წახშირბადი ჯგან.
ka	ანრი რუსო გარდაიცვალა პარიზი -ში.
	კატალონია მდებარეობს ესპანეთი -ში.
	바하칼리포르니아주는 캘리포니아주와 (과) 국경을 공유합니다.
ko	레오 10세의 위치는 교황입니다.
	캐나다는 이탈리아와의 외교 관계를 유지합니다.
	Prophetia Michaeae est pars Biblia.
la	Carentonium Cum shares terminus Lutetia.
	Carolus Bildt in communicate ad lingua Suecica.
	Pensilvanija dalijasi riba su Merilandas.
lt	Viskonsinas sostinė yra Madisonas.
	Park Chan-wook naudojamas bendrauti Korėjiečių kalba.
	Altaja Republika oficiālā valoda ir krievu valoda.
lv	Francs Kafka dzimtā valoda ir vācu valoda.
	Audi Allroad Quattro ražo Audi.
	Tour de France dinamakan Perancis.
ms	Goa berkongsi sempadan dengan Maharashtra.
	Modal Amerika British ialah London.
	allmennaksjeselskap is een juridische term in Noorwegen.
nl	San Marinese voetbalbond is lid van FIFA.
	De oorspronkelijke taal van Nouvelle Star is Frans.
	Armand Marrast pracował w Paryż.
pl	Irlandia nosi imię Irlandia.
	Oficjalnym językiem Kanton Jura jest język francuski.
	Denver e Nairóbi são cidades gêmeas.
pt	Arábia Saudita mantém relações diplomáticas com México.
	Nyepi está localizado em Bali.
	Districtul Darnah este localizat în Libia.
ro	Sediul central al Toyota este în Toyota.
	Roy Orbison folosit pentru a comunica în limba engleză.
	Венанций Фортунат имеет позицию епископ.
ru	Renault 21 производится Renault.
	Пиччотто, Ги играет гитара.
	Austrália udržiava diplomatické vzťahy s Nórsko.
sk	Alanín pozostáva z dusík.
	Optický ďalekohľad je podtrieda teleskop.
	Fergus Morton se je rodil v Glasgow.
sl	Nemčija je članica NATO.
	Cenk Renda se je rodil v Turčija.
	Nic Chagall interpreton Trance muzikë.
sq	Marie Liljedahl është Suedia qytetar.
	Pallati i Fontainebleau është në pronësi të Francë.
	Нортхемптоншир дели границу са Бакингемшир.
sr	периапсис је део орбита.
	Патрик Дотерс је рођен у Беркли.
	Det officiella språket för Savukoski är finska.
sv	The Upsetters spelar reggae musik.
	Arakidonsyra består av kol.
	வெனிசுவேலா செருமனி உடன் இராஜதந்திர உறவுகளைப் பேணுகிறது.
ta	தையால் கந்தகம் ஐக் கொண்டுள்ளது.
	எக்சு.கோட் ஆப்பிள் நிறுவனம் ஆல் உருவாக்கப்பட்டது.
	ประเทศซูดาน อยู่ใน ทวีปแอฟริกา
th	ภาษาราชการของ รัฐไอโอวา คือ ภาษาอังกฤษ
	กรุงเทพมหานคร เป็นเมืองหลวงของ ประเทศไทย
	Markus Feldmann Bern 'da çalışırdı.
tr	Graduate Institute of International and Development Studies şirketinin genel merkezi Cenevre dedir.
	Afganistan Demokratik Cumhuriyeti 'un başkenti Kâbil' dir.
	Столиця Сирія - Дамаск.
uk	Комплекс Наполеона названий на честь Наполеон I Бонапарт.
	Нижня Канада ділиться межею з Вермонт.
	مملکت سربیا کا دارالحکومت بلغراد ہے۔
ur	میکسیکو قومی فٹ بال ٹیم فیفا کا ممبر ہے۔
	یورپی اتحاد بیلاروس کے ساتھ سفارتی تعلقات برقرار رکھے ہوئے ہے۔
	Ngôn ngữ chính thức của Lampung là tiếng Indonesia.
vi	Ngôn ngữ chính thức của Viitasaari là tiếng Phần Lan.
	Ả Rập Saudi duy trì quan hệ ngoại giao với Yemen.
	蒙马特圣伯多祿堂以西門彼得命名。
zh	Sun Media集團的总部位于多伦多中。
	多米尼克·杜卡曾经在布拉格中工作。

Figure 5: Data samples continued.

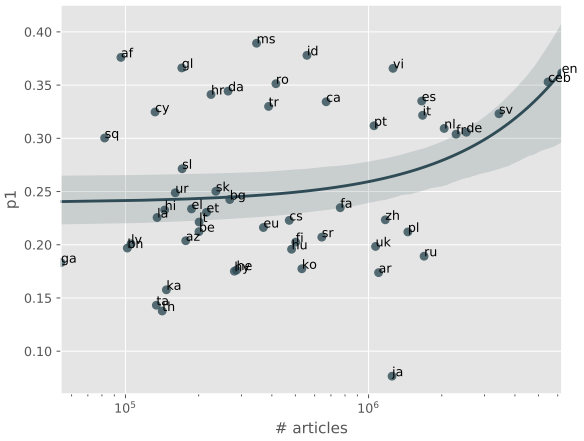


Figure 6: Scatter plot of p1 TyQ and number of articles in the corresponding Wikipedia. There is no clear trend visible.

Bibliography

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively multilingual word embeddings. *Computing Research Repository*, arXiv:1602.01925.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

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, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020a. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020b. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7375–7388. Association for Computational Linguistics.

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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Computing Research Repository*, arXiv:2005.14165.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Aditi Chaudhary, Karthik Raman, Krishna Srinivasan, and Jiecao Chen. 2020. DICT-MLM: improved multilingual pre-training using bilingual dictionaries. *Computing Research Repository*, arXiv:2010.12566.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. 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. Association for Computational Linguistics.

Jonathan H. Clark, Jennimaria Palomaki, Vitaly Nikolaev, Eunsol Choi, Dan Garrette, Michael Collins, and Tom Kwiatkowski. 2020a. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020b. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12:2493–2537.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. 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. 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 7059–7069. Curran Associates, Inc.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Zhen Cui, Hong Chang, Shiguang Shan, and Xilin Chen. 2014. Generalized unsupervised manifold alignment. In *Advances in Neural Information Processing Systems*, volume 27, pages 2429–2437. Curran Associates, Inc.

Andrew M. Dai, Christopher Olah, and Quoc V. Le. 2015. Document embedding with paragraph vectors. *Computing Research Repository*, arXiv:1507.07998. NIPS Deep Learning Workshop.

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 and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

David M. Eberhard, F. Simons Gary, and D. Fennig (eds.) Charles. 2020. *Ethnologue: Languages of the World*, 23rd edition. SIL International.

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

J. R. Firth. 1957. A synopsis of linguistic theory 1930-55. In *Studies in Linguistic Analysis (special volume of the Philological Society)*, volume 1952-59, pages 1–32, Oxford. The Philological Society.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 710–721. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

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

Geoffrey E. Hinton, James L. McClelland, and David E. Rumelhart. 1990. Distributed representations. In *The Philosophy of Artificial Intelligence*, Oxford readings in philosophy, pages 248–280. Oxford University Press.

Geoffrey E Hinton et al. 1986. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, Lawrence Erlbaum Associates.

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

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *Computing Research Repository*, arXiv:2003.11080.

W. John Hutchins. 2004. The Georgetown-IBM experiment demonstrated in january 1954. In *Machine Translation: From Real Users to Research, 6th Conference of the Association for Machine Translation in the Americas, AMTA 2004,*

Washington, DC, USA, September 28-October 2, 2004, Proceedings, volume 3265 of *Lecture Notes in Computer Science*, pages 102–114. Springer.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *Proceedings of the 10th Machine Translation Summit (MT Summit), 2005*, pages 79–86.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4483–4499. Association for Computational Linguistics.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1188–1196. JMLR.org.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jindrich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 1663–1674. Association for Computational Linguistics.

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. *Computing Research Repository*, arXiv:1907.11692.

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-2014)*, pages 3158–3163, Reykjavik, Iceland. European Languages Resources Association (ELRA).

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6294–6305. Curran Associates, Inc.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *Computing Research Repository*, arXiv:1309.4168.

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

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis M. Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4034–4043. European Language Resources Association.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 1-8, 2000*, pages 440–447. ACL.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher 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, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.

Matthew 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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4996–5001. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 2362–2376. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *Technical report, OpenAI*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 151–164. Association for Computational Linguistics.

Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. 2020. LAReQA: Language-agnostic answer retrieval from a multilingual pool. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5919–5930. Association for Computational Linguistics.

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

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 5149–5152. IEEE.

Hinrich Schütze. 1992a. Dimensions of meaning. In *Proceedings Supercomputing '92, Minneapolis, MN, USA, November 16-20, 1992*, pages 787–796. IEEE Computer Society.

Hinrich Schütze. 1992b. Word space. In *Advances in Neural Information Processing Systems 5, NIPS Conference, Denver, Colorado, USA, November 30 - December 3, 1992*, pages 895–902. Morgan Kaufmann.

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, Melbourne, Australia. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, pages 194–197. ISCA.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147. Association for Computational Linguistics.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, Lisbon, Portugal. Association for Computational Linguistics.

Alan Turing. 1950. Computing machinery and intelligence. *Mind*, LIX(236):433–460.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4406–4417. Association for Computational Linguistics.

Ivan Vulić 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 and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 719–725, Beijing, China. Association for Computational Linguistics.

Ivan Vulić, Sebastian Ruder, and Anders Søgaard. 2020. Are all good word vector spaces isomorphic? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3178–3192. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, pages 3266–3280. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Chang Wang and Sridhar Mahadevan. 2009. Manifold alignment without correspondence. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1273–1278.

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, Denver, Colorado. Association for Computational Linguistics.

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 60–67, Vancouver, Canada. Association for Computational Linguistics.