

# Sigmorphon 2019 Task 2 system description paper: Morphological analysis in context for many languages, with supervision from only a few

Jared Kelly\* Brad Aiken\*\* Alexis Palmer\*  
Suleyman Olcay Polat\*\* Taraka Rama\* Rodney Nielsen\*\*

\*Department of Linguistics, University of North Texas

{jared.kelly, alexis.palmer}@unt.edu, tarakark@ifi.uio.no

\*\*Department of Computer Science and Engineering, University of North Texas

{bradfordaiken, suleymanolcaypolat}@my.unt.edu, rodney.nielsen@unt.edu

## Abstract

This paper presents the UNT HiLT+Ling system for the Sigmorphon 2019 shared Task 2: Morphological Analysis and Lemmatization in Context. Our core approach focuses on the morphological tagging task; part-of-speech tagging and lemmatization are treated as secondary tasks. Given the highly multilingual nature of the task, we propose an approach which makes minimal use of the supplied training data, in order to be extensible to languages without labeled training data for the morphological analysis task. Specifically, we use a parallel Bible corpus to align contextual embeddings at the verse level. The aligned verses are used to build cross-language translation matrices, which in turn are used to map between embedding spaces for the various languages. Finally, we use sets of inflected forms, primarily from a high-resource language, to induce vector representations for individual UniMorph tags. Morphological tagging is performed by matching vector representations to embeddings for individual tokens. While our system results are dramatically below the average system submitted for the shared task evaluation campaign, our method is (we suspect) unique in its minimal reliance on labeled training data.

## 1 Introduction

This paper describes the UNT HiLT+Ling system submission for the Sigmorphon shared task on morphological analysis and lemmatization in context (McCarthy et al., 2019). We focus primarily on the morphological tagging task, treating part-of-speech tagging and lemmatization as secondary tasks. We approach morphological analysis from the perspective of low-resource languages, aiming to develop an approach which exploits existing language resources in order to make morphological analysis in context feasible for languages

without annotated training data. We propose a model to perform morphosyntactic annotation for any language with a translation of the Bible. According to Wycliffe<sup>1</sup>, there are currently 683 languages in the world which contain a translation of the entire Bible, and an additional 1534 languages for which the entire New Testament, and sometimes other sections, are available.

We train contextual word representations using ELMo (Peters et al., 2018) and align embedding spaces for language pairs using Bible verse numbers as an alignment signal. We then compute vector representations for UniMorph tags in English and project those representations into the target language. The projected morpheme tag embeddings are used to identify morphological features and label tokens in context with UniMorph tags.

We give a system overview in Section 2, with more detailed model descriptions in Section 5. The system’s performance is currently poor; we outline known limitations and make some suggestions for improvement.

## 2 System Overview

The system we developed for Sigmorphon 2019 Task 2 can be divided into two parts: the core model and two additional non-core components. The core model is responsible for the morphological tagging task, our main focus. The two non-core components are part-of-speech tagging and lemmatization.

**Core model: Minimally-supervised morphological analysis in context.** Following task specifications, we aim to predict UniMorph tags for words in context. Our approach is designed to work on new languages with minimal supervision. Specifically, the base model uses the following forms of supervision: a) multilingual bible

<sup>1</sup><http://www.wycliffe.net/statistics>

data, verse-aligned; and b) roughly twenty words per from the training data per UniMorph tag. Once this model has been developed, it can be applied for a new language with no annotated training data for the task; the only data needed is a Bible in that language.

The steps in the process (explained in detail in Section 5.1) are as follows:

1. Learn sentence-level ELMO embeddings (Peters et al., 2018) for each language.
2. Use verse-aligned data to learn a vector translation matrix (following Mikolov et al., 2013a) between each language and English.
3. Compute a vector representation for each UniMorph tag.
4. For UniMorph tags found in English, map tag vectors into the other languages which use the tag, by way of the relevant translation matrix. For tags not found in English, compute vector representations for each tag in the language-specific space.
5. Identify all UniMorph tags represented in the embedding for a given word, treating morphological analysis in the style of analogy tasks (Mikolov et al., 2013b).

**POS tagging and lemmatization.** POS tagging and lemmatization are treated as non-core components of the model. In other words, we incorporate these tasks into our model in order to meet the requirements of the competition. For these two tasks, greater supervision is allowed, and models are learned from the training data provided. The POS tagger in our system is a straightforward HMM model, and lemmatization is done with a seq2seq neural architecture. See Section 5.2 for more detailed descriptions of the models.

### 3 Related Work

The core idea of using the Bible as parallel data in low-resource settings is largely inspired by previous work. The Bible has been used as a means of alignment for cross-lingual projection, both for POS tagging (Agic et al., 2015) and for dependency parsing (Agic et al., 2016), as well as for base noun-phrase bracketing, named-entity tagging, and morphological analysis (Yarowsky et al., 2001) with promising results.

Peters et al. (2018) introduce ELMo embeddings, contextual word embeddings which incorporate character-level information using a CNN.

Both of these properties - sensitivity to context and the ability to capture sub-word information - make contextual embeddings suitable for the task at hand.

In order to make embeddings useful across languages, we need a method for aligning embedding spaces across languages. Ruder et al. (2017) provide an excellent survey of methods for aligning embedding spaces. Mikolov et al. (2013a) introduce a translation matrix for aligning embedding spaces in different languages and show how this is useful for machine translation purposes. We adopt this approach to do alignment at the verse level. Alignment with contextual embeddings is more complicated, since the embeddings are dynamic by their very nature (different across different contexts). In order to align these dynamic embeddings, Schuster et al. (2019) introduce a number of methods, however they all require either a supervised dictionary for each language, or access to the MUSE framework for alignment, neither of which we assume in our work.

The UniMorph 2.0 data-set (Kirov et al., 2018) provides resources for morphosyntactic analysis across 111 different languages. The work described here uses the tag set from UniMorph.

## 4 Data

This section describes the data resources used for training and evaluating the system.

### 4.1 Bible data

The main data used for building our core model is a multilingual Bible corpus. For as many of the shared task languages as possible (41), we use the corpus from Christodouloupoulos and Steedman (2015). Bibles for an additional 19 languages were sourced elsewhere. Of the remaining 11 languages, we use proxy languages (Section 4.2) for 9. For two languages (Akkadian and Sanskrit), we were unable to locate a suitable Bible in time. Where there are multiple data sets for a given language, we use the same Bible for all data sets.

For some languages we have access to the entire Bible, and for others only the New Testament (NT). This introduces discrepancies in the amount of data used to train embeddings from language to language, as the Old Testament is much longer than the New Testament.

The Bible is a natural source of parallel data, as it is available (either in whole or in parts) in

| Shared task language    | ISO code | Proxy language   | ISO code |
|-------------------------|----------|------------------|----------|
| UD_Belarusian-HSE       | bel      | Russian          | rus      |
| UD_Breton-KEB           | bre      | Irish Gaelic     | gle      |
| UD_Galician-CTG         | glg      | Portuguese       | por      |
| UD_Galician-TreeGal     | glg      | Portuguese       | por      |
| UD_Gothic-PROIEL        | got      | Icelandic        | isl      |
| UD_Norwegian-Nynorsk    | nno      | Icelandic        | isl      |
| UD_Norwegian-NynorskLIA | nno      | Icelandic        | isl      |
| UD_Upper_Sorbian-UFAL   | hsb      | Czech            | ces      |
|                         |          | <b>Dialect</b>   |          |
| UD_Armenian-ArmTDP      | hy       | Eastern Armenian | hye      |
| UD_Irish-IDT            | ga       | Irish Gaelic     | gle      |
| UD_Persian-Seraji       | fa       | Western Persian  | pes      |
|                         |          | <b>No Bible</b>  |          |
| UD_Akkadian-PISANDUB    | akk      |                  |          |
| UD_Sanskrit-UFAL        | san      |                  |          |

Table 1: Shared task languages for which a proxy language or close dialect was used, and languages for which no Bible was used.

over one thousand languages, including many low-resource languages. One advantage of using the Bible, beyond its wide availability in translation for free, is that its verses are fairly well-aligned in meaning across languages (unlike words or even sentences). One drawback to using Bible data is the archaic nature of the language. For example, even if we use a modern translation, the English Bible contains fewer than 15,000 different word types, and no occurrences of modern words (e.g. *Republican*, *computer*, or *NASA*).

The limited domain of the text offers both advantages and disadvantages. On the one hand, much of the vocabulary found in the shared task evaluation data does not occur in the Bible. Using embeddings trained on the Bible, then, results in an extremely large number of out-of-vocabulary tokens at test time. On the other, the semantic territory covered by the embedding spaces varies remarkably little from language to language, increasing the feasibility of aligning embedding spaces across multiple languages.

## 4.2 Proxy languages

In order to do morphological analysis for a given language, our method requires access to a digitally-available version of at least portions of the Bible for that language. At the time the model was developed, we did not have access to Bibles for all shared task languages. For each missing

language, we select a proxy language (Table 1). For example, we don’t have a Bible for Galician, so at every stage in the process where the Galician Bible would be used, we substitute the Portuguese Bible, treating Portuguese as pseudo-Galician. We identify two different cases of proxy language substitution. In some cases, we are able to select a closely-related dialect for the target language. In others, the proxy language is selected based on a combination of morphological similarity (typologically speaking) and language relatedness.<sup>2</sup>

## 4.3 Sigmorphon data

We use the provided training data (McCarthy et al., 2018) primarily to train a part-of-speech tagger and lemmatizer for each shared task data set, and the provided test data is used to evaluate the system. We use portions of the training data for three other purposes: a) to build contrasting sets of words for each UniMorph tag (Section 5.1.3); b) to build lists of UniMorph tags relevant for each language; and c) to create a simple baseline for the two languages for which we have no Bible, proxy language or otherwise.

<sup>2</sup>In an early experiment, we investigated the effectiveness of similarity measures over language vectors (Malaviya et al., 2017; Littell et al., 2017) for selecting proxy languages. The results were mixed, so we opted for expert selection of proxy languages instead. Lin et al. (2019) discusses some of the issues involved.

## 5 Models

The model description consists of two parts: the core model, for morphological analysis, and two non-core components, for part-of-speech tagging and lemmatization.

### 5.1 Core model: morphological analysis

Our core system addresses the task of morphological analysis with minimal supervision from labeled training data. The approach exploits parallel data in the form of a multilingual Bible corpus.

#### 5.1.1 Contextual embeddings for every Bible

Prior research has shown embedded word vector representations are capable of capturing contextual nuances in meaning beyond one sense per word (Arora et al., 2018, for example). Because context variance is an important factor affecting morphological analysis, we use ELMo embeddings (Peters et al., 2018) as our base representation. As a first step, we train separate ELMo models on each of the Bible translations in our corpus. For each language, we hold out four books (Mark, Ephesians, 2 Timothy, and Hebrews) for model evaluation and train on all remaining books. Models are trained at the sentence level, using default parameter settings and following recommendations from the AllenNLP `bilm-tf` repository.<sup>3</sup>

#### 5.1.2 Verse alignment for embedding projection

The next step is to use the natural verse alignment of the Bible to learn projections from one embedding space to another, treating English as the source language and learning projections into the embedding spaces for each of our non-English Bible languages in turn.

Mikolov et al. (2013a) show that type-level embedding spaces (e.g. `word2vec`) can be projected across languages by calculating a translation matrix from a set of type-level translation word pairs. The translation matrix is a vector of dimension-wise factors by which word representations from a source language can be multiplied to transform them into parallel word representations in the target language embedding space.

Aligning contextual representations such as ELMo is more complicated, as there is no good way of aligning words between two language embedding spaces without a dictionary and without

losing the encoded information about contextual polysemy, for which ELMo is particularly useful.

Schuster et al. (2019) propose using context-free anchors to align contextually-dependent embedding spaces (such as ELMo). We propose instead to calculate translation matrices at the verse level, computing the representation for each verse as the unweighted average of its constituent contextual word embeddings.

First, we compute ELMo embeddings for each token in a small subset of the Bible: Psalms (OT) and Romans (NT). For a given language pair, we compute a verse embedding for each verse that appears in both Bibles (some verses are missing in some languages, and some languages have extra verses)<sup>4</sup> and derive the translation matrix for that language pair using the standard method, as introduced by Mikolov et al. (2013a).

Given pairs of verse vectors in a source and target language  $\{x_i, z_i\}_{i=1}^n$  respectively, we calculate the translation matrix ( $W$ ) between the two languages utilizing gradient descent, as follows:

$$\min_W \sum_{i=1}^n \| Wx_i - z_i \|^2$$

#### 5.1.3 Inducing vectors for UniMorph tags

In lieu of using supervised, annotated data for training the model with morphological information, we work from the hypotheses that each of the 42 UniMorph tags can be isolated in the embedding space and that we can derive a vector representation for each tag, applying a process similar to the well-known analogy tasks of Mikolov et al. (2013b). For this purpose, we build small hand-curated data sets (only in English), with contrasting sets of words for each tag. In other words, for each UniMorph tag found in English, we collect from the training data one set of words with the tag and a parallel set without it. The word sets do not necessarily contain minimal pairs, but rather groups of words that are matched for part-of-speech. For example, for the plural tag PL, we build a list of 10 plural tokens (e.g. [*women, cats, dogs, deer, ...*]) and another list of 10 singular tokens (e.g. [*man, car, dog, apple, ...*]). The (vectors for) the set of words with the tag are subtracted

<sup>4</sup>Even though the differences of diverse and distant languages result in occasional discrepancies in the verse alignment, we believe that verse-level alignment offers closer semantic matching than unsupervised sentence-level alignment could achieve. Across the 60 languages for which we have Bibles, the average ratio of sentences to verses is 1.27 to 1.

<sup>3</sup><https://github.com/allenai/bilm-tf>

from (vectors for) the set of words without the tag. More precisely, we take the weighted average of both sets of words, in which those with the tag are weighted 1, and those without it are weighted -1.

Having derived a vector representation for each UniMorph tag, these vectors can now be projected from English into the target language using the respective translation matrix. Rather than projecting every tag into every language, we project only the tags that are seen in a given language’s training data.

Of course, only a subset of all UniMorph tags are found in English. For those which do not appear in the English data (e.g. Ergative), an additional method was developed using the Sigmorphon training data in other languages. When tagging a language that has the tag ERG in the training data, we build new word list pairs specific to that language and calculate the UniMorph tag representation as described above.

#### 5.1.4 Morphological analysis

To assign UniMorph tags to words at test time, a sequence of tokens in context (one sentence at a time) is fed into ELMo using the target language ELMo model, generating contextual embeddings for each word in the sequence. Next, for each token, we iteratively subtract each of the target language’s possible UniMorph vectors and search for another word in the target language whose embedding is within 0.1 cosine distance of the resulting vector. For example, when tagging the German word *Kinder* (children), subtracting the vector representation for the Plural tag should result in a vector that is close to that for *Kind* (child). This subtraction process is applied to every word, for every UniMorph tag found in the language. Whenever a word is found within the threshold of the derived embedding, the tag that resulted in the successful transformation is assigned to that token. In the example above, *Kinder* gets tagged with PL.

Intuitively, this method is plausible because words, their inflected forms, synonyms, and closely related terms tend to occur in tight clusters in embedding spaces. Therefore, subtracting the embedding for the PL tag from the embedding for *the* should not produce a close match in English, since the plural tag is never associated with *the*. This would not be a grammatically meaningful transformation.

#### 5.1.5 Baselines

We use two different baselines for the morphological analysis task.

**No-embedding baseline.** This method is used to tag the two languages for which we have no Bible, not even for a proxy language, and thus have no Bible-trained word embeddings for the language. Under this approach, each word is simply labeled with all tags it has been seen with in the training data.

**Embedding baseline.** This method makes use of the verse embeddings described above and was deployed to do tagging where time constraints prohibited implementation of the full model for a given language.

The contextualized word representations built to support the embedding projection process are collected into a set of dictionaries (one for each language) of seen tokens and their associated vectors. In this setting, instead of re-training the ELMo model on test data in context, we retrieve stored vectors for tokens to be tagged. This method has clear shortcomings, both with respect to coverage of the model and regarding the handling of polysemous tokens.

#### 5.2 Non-core components: POS tagging and lemma generation

For part-of-speech tagging, we implement a Hidden Markov Model Viterbi algorithm trained on the Sigmorphon training and development datasets. Given our interest in methods which reduce the need for large labeled corpora and supervised learning, we additionally implemented some simple heuristics based on previously-generated morpheme tags.

For example, a word is given a higher probability of being tagged as a verb if it has a modal, tense, or other conjugative tag already assigned to it (e.g., V.PTCP or PRS). These heuristics were designed to be entirely language-neutral, generalizing to the full set of test languages.

As a final task, we perform lemma generation using a joint neural model following [Malaviya et al. \(2019\)](#)’s proposed method. The joint model consists of a simple LSTM-based tagger to recover the morphology of a sentence and a sequence-to-sequence model with hard attention mechanism as a lemmatizer. The lemmatization model trains over words and their morphological information

|        | Lemma Accuracy | Lemma Levenshtein | Morph Accuracy | Morph F1 |
|--------|----------------|-------------------|----------------|----------|
| Mean   | 83.143         | 0.5511            | 15.689         | 51.870   |
| Median | 90.66          | 0.16              | 13.98          | 55.18    |

Table 2: Shared task results for our models, across all languages.

|  | Morph Accuracy | Morph F1 |
|--|----------------|----------|
| Full Bible (n=77)                                | 16.08          | 52.64    |
| NT only (n=17)                                   | 13.12          | 48.68    |
| Bible from close dialect (n=3; 2 NT, 1 full)     | 20.55          | 57.06    |
| Bible from proxy language, (n=8; all full Bible) | 12.88          | 51.57    |
| No Bible   | 26.6           | 42.23    |

Table 3: Morphological analysis results. Macro-average for subgroups of data sets, categorized according to resources available for embedding training, alignment, and projection.

recovered with the tagger. To counter exposure bias, all training is done with Jackknifing.

### 5.3 Limitations - there are many

The models as described above are subject to many limitations, and we have many ideas for improving the system.

First, the model is computationally intensive and time intensive, to an extent that meant we only applied the full model to a fraction of the data.

Because producing ELMo embeddings on-the-fly is so time consuming, we took some shortcuts in order to get results in time for submission. Word types already tagged were stored together with their tags after the first encounter, and the tags retrieved for later occurrences. Also, only a subset of the test sentences were in fact tagged with the ELMo approach at all. These two things together resulted in many false positives and redundant tags (e.g. the same noun tagged as both nominative and accusative). We feel confident that a full run of the system, however long it takes, will result in much better performance.

Second, our method for tagging words with UniMorph tags does nothing to constrain the set of possible tags, allowing multiple conflicting tags to be simultaneously assigned. Application of output constraints could go a long way toward solving this issue.

Third, we would like to rework our method for collecting pairs of word lists for derivation of vector representations for UniMorph tags. A problem with the current method is that it assumes the existence of inflected/non-inflected word pairs for all tags, and in all languages. In fact, many morphological paradigms do not consist of contrasts be-

tween inflected and un-inflected forms (these are perhaps more common in English than in most languages), but rather of sets of inflectional options, one of which is likely to occur. Our model does not currently account well for this aspect of morphology.

For example, when tagging the German article *dem* (definite, masculine, **dative**), subtracting the vector representation for the Dative tag under our current model results in an ill-defined form; there is no article that is definite and masculine and with undefined case. Instead, we would like for the process to yield a set of vectors, close to those for the articles *der* (definite, masculine, **nominative**); *den* (definite, masculine, **accusative**); and *des* (definite, masculine, **genitive**).

Fourth, the system is very bad at handling morphological analysis for out-of-vocabulary tokens, and there are many out-of-vocabulary tokens.

## 6 Results

Table 2 provides an overview of our system results. Additional discussion of results can be found in McCarthy et al. (2019). The results are uncontroversially bad, particularly for the morphological analysis task. For this portion of the task, our accuracies are dramatically lower than all other teams (at least 50% worse than every other team, on most languages). Some of this performance gap surely can be attributed to the fact that we make very minimal use of the training data supplied, but not all of it! We strongly believe that the limitations described in Section 5.3 have severely decreased our results, and we look forward to giving our method a true test in the near

future.

For lemmatization, we come closer to average performance, coming in at roughly 12 percent less accurate on average (across languages) than the top-performing submitted system.

Table 3 looks at results compared to the amount and type of Bible data used to train embeddings for each language. Performance suffers when training on only the New Testament, compared to the full Bible. Surprisingly, proxy language training shows only a slightly lower average performance compared to training and testing on the same language. Of course, all results need to be interpreted with respect to the limitations previously discussed.

## 7 Discussion

In addition to the model and implementation limitations discussed in Section 5.3, there are a number of extensions which could be considered for improving the model.

Our current model allows a mismatch between granularity for training of the embedding spaces (sentences) and granularity for alignment of the embedding spaces (verses). We'd like to experiment with verse-trained models as well.

We would also like to train on all of our Bible data, without holding out any data for evaluation of the embedding space (i.e. the four books mentioned in Section 4). For languages for which we don't have a Bible, we will investigate new methods for identifying transfer languages (Lin et al., 2019).

Even though our models as implemented prior to submission failed to attain reasonable accuracy on the morphological analysis task, we believe that performance can be improved and that the general architecture deserves further exploration. Ideally, our model could extend to any of the 800 (or more) languages that has a translation of the entire bible, opening new frontiers for minimally-supervised morphological analysis.

## Acknowledgments

Thanks to the reviewers for helpful feedback. Computational resources were provided by UNT office of High-Performance Computing.

## References

- Zeljko Agic, Dirk Hovy, and Anders Sgaard. 2015. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 268–272.
- Zeljko Agic, Anders Johannsen, Barbara Plank, Hector Martinez Alonso, Natalie Schluter, and Anders Sgaard. 2016. [Multilingual Projection for Parsing Truly Low-Resource Languages](#). *Transactions of the Association for Computational Linguistics*, 4:301–312.
- Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association of Computational Linguistics*, 6:483–495.
- Christos Christodouloupoulos and Mark Steedman. 2015. [A massively parallel corpus: the Bible in 100 languages](#). *Language Resources and Evaluation*, 49(2):375–395.
- Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian J. Mielke, Arya D. McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. [UniMorph 2.0: Universal Morphology](#). *CoRR*, abs/1810.11101.
- Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of ACL 2019*.
- Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 8–14.
- Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark.
- Chaitanya Malaviya, Shijie Wu, and Ryan Cotterell. 2019. [A simple joint model for improved contextual neural lemmatization](#). *CoRR*, abs/1904.02306.
- Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2018. [Marrying Universal Dependencies and Universal Morphology](#). In *Proceedings of the Second Workshop*

on *Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium. Association for Computational Linguistics.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Crosslinguality and context in morphology. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Florence, Italy. Association for Computational Linguistics.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. [Exploiting Similarities among Languages for Machine Translation](#). *CoRR*, abs/1309.4168.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. 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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). *CoRR*, abs/1802.05365.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2017. [A survey of cross-lingual embedding models](#). *CoRR*, abs/1706.04902.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. [Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing](#). *CoRR*, abs/1902.09492.

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, HLT '01*, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.