Proceedings of the 15th Workshop on Building and Using Comparable Corpora (BUCC 2022) @LREC2022 , pages 43–49
Marseille, June 25, 2022

CUNI Submission to the BUCC 2022 Shared Task on Bilingual Term Alignment

Borek Požár, Klára Tauchmanová, Kristýna Neumannová, Ivana Kvapilíková and Ondřej Bojar
Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics, Charles University,
Prague, Czech Republic,
pozar.borek@gmail.com, klara.tauchmanova@gmail.com, kristynaneumannova@gmail.com,
kvapilikova@ufal.mff.cuni.cz, bojar@ufal.mff.cuni.cz

Abstract

We present our submission to the BUCC Shared Task on bilingual term alignment in comparable specialized corpora. We devised three approaches using static embeddings with post-hoc alignment, the Monoses pipeline for unsupervised phrase-based machine translation, and contextualized multilingual embeddings. We show that contextualized embeddings from pretrained multilingual models lead to similar results as static embeddings but further improvement can be achieved by task-specific fine-tuning. Retrieving term pairs from the running phrase tables of the Monoses systems can match this enhanced performance and leads to an average precision of 0.88 on the train set.

Keywords: word embeddings, usupervised learning, alignment, multiword expressions, embedding mapping

1. Introduction

The goal of the task is to find equivalent expressions (both one- and multi-word, we will call them *terms*) in two languages. The inputs are comparable corpora C_1 and C_2 in languages L_1, L_2 and lists of terms D_1, D_2 which are to be mapped onto each other, where D_1 is extracted from C_1 and D_2 from C_2. The output $O_{1,2}$ is supposed to be a list of term pairs t_1, t_2 that are translations of each other, where $t_i \in D_i \in L_i$. The output list should be ordered based on decreasing confidence in the translation. Some terms from D_1 may not have a translation in D_2, some may have multiple translations, and conversely. The submission length is limited to $10 \cdot \frac{D_1+D_2}{2}$. For the training, the gold output $G_{1,2}$ is available. Average Precision is used as a metric and the usage of any additional data (except the CCAligned corpus (El-Kishky et al., 2020), from which the datasets were extracted) is allowed.

The training language pair was English-French, the test datasets were supposed to contain three language pairs – English-French, English-German and English-Russian, however only the English-French was released.

We took three different approaches to find the candidate translation pairs. In the first approach, we create static FastText term embeddings, align them and then search for the nearest neighbours in the embedding space (section 3). The second approach uses an unsupervised phrase-based machine translation (MT) system Monoses and searches in its translation tables (section 4). We also experiment with their combination (section 5). The third approach is similar to the first one, but the embeddings are extracted from pretrained multilingual language models (section 6).

2. Related Work

The task of bilingual term alignment is close to the task of bilingual lexicon induction (BLI) which was initially tackled by statistical deciphering (Ravi and Knight, 2011). Later works on BLI are mostly embedding-based, where authors generate two monolingual embedding spaces and align them post-hoc either with the supervision of an existing lexicon (Mikolov et al., 2013) or with a very weak supervision of identical strings and numerals (Artetxe et al., 2017) or no supervision at all (Conneau et al., 2018; Artetxe et al., 2018a). The cross-lingual embedding space is then searched for the nearest neighbours.

Alternatively, Artetxe et al. (2019a) use cross-lingual embeddings to build a phrase-table of an unsupervised statistical MT system which is used to generate a synthetic parallel corpus. The bilingual lexicon is extracted from the synthetic corpus by using statistical word alignment techniques. Shi et al. (2021) combine unsupervised bitext mining and unsupervised word alignment to obtain a lexicon of state-of-the-art quality.

3. First Approach

3.1. Data Preprocessing

Since the first approach is based on the embedding mapping, we needed to merge the multiword expressions to obtain their embeddings. We replaced the spaces in such expressions by underscores, so when splitting on whitespaces, they were treated as one word. In order to replace the right spaces, we needed to find the multiword expressions in the corpus. Since many of the words were in an inflected form, we used lemmatization. We tried UDPipe 1 (Straka and Straková, 2017)

and UDPipe 2 (Straka, 2018), both trained on Universal Dependencies data. The former one was faster (several hours on 1 CPU thread) than the latter one (several hours on GTX1080 GPU), but produced worse results, as expected. All the characters in the corpora were changed to lowercase and numbers were normalized into a `<num>` token, since their meaning is not important for the task and normalization helps the embeddings training.

Some sentences contained more than one of the term from the list and some of the terms were overlapping, for example the English term list contained terms `valid email`, `email address` and `valid email address`. In order to deal with this, we added all possible versions of the sentence to the corpus, including the original one. That way, the embeddings could be trained for all the terms, which we consider correct, because they all appeared in the sentence. An example original sentence from the corpus `please enter a valid email address` with an example term list `valid email, email address, valid email address` would then appear in our training data in four variants:

- `please enter a valid_email address`
- `please enter a valid_email_address`
- `please enter a valid email_address`
- `please enter a valid email address`

We also needed to lemmatize the term lists. We tried passing the term lists right into the lemmatizer, but that produced very bad results, presumably because the lemmatizers work with a sentence context. Therefore, for each term, we looked at how it was lemmatized in the corpus and used the most frequent lemma. Then after retrieving the translations at the end of this approach, we have converted the lemmatized terms back to their original versions in order to produce the correct output.

3.2. Cross-lingual Word Embeddings

We used an unsupervised method provided by FastText (Bojanowski et al., 2017) to train word vectors for both preprocessed monolingual corpora. Monolingual embeddings of dimension 300 were learned using the skip-gram model with subwords formed from 3 to 6 substring characters.

The obtained monolingual word embeddings were then aligned into a common space using an unsupervised method provided by the MUSE tool (Conneau et al., 2018). The unsupervised method leverages adversarial training to learn a linear mapping from the source to the target space. The training was run for 5 epochs with 1,000,000 iterations per epoch.

3.3. Resulting Term Dictionary

The resulting dictionary was created by computing neighbours of individual terms from the given list of terms. For each term from the source language, we compute k nearest neighbours (for $k = 1, 2, 3, 5, 10$) in the target language. Similarly, we computed k nearest neighbours for each term from the target language. For each translation, we considered the spatial similarity as a score, summing it if we found given translation pair when searching both directions. Then we filter out pairs that contain terms not included in the given list of terms (both from the source and the target language).

Finally, we arranged pairs of terms into the resulting dictionary in the following order: first the nearest neighbour for each source term in an alphabetical order, then the second nearest neighbour, etc. If a source term no longer had another neighbour, it was skipped. Table 1 contains first 30 translations from the train set. The results of this approach are presented in Table 4.

abreast	affût
absence	absence
absolute	absolue
absolute freedom	liberté absolue
absolutely	absolument
academic	académique
acceptable	acceptable
access control	système de contrôle
accessible	accessible
accident	accident
account	compte
account number	numéro de compte
accurate	précises
acid	acide
acoustic	acoustique
action	action
active	actifs
active life	vie active
actively	activement
active members	membres actifs
activists	activistes
activities	activités
actors	actrice
adaptation	adaptation
additional	supplémentaires
additional charge	charge supplémentaire
additional cost	coût supplémentaire
additional income	revenu supplémentaire
additional info	informations supplémentaires
additional information	informations supplémentaires

Table 1: First 30 translations on train set from the first approach.

4. Second Approach

The second approach is based on unsupervised phrase-based machine translation. The model was trained using only the given comparable corpora. As the main

component of the pipeline, we used the Monoses tool (Artetxe et al., 2019b). The tool processed raw corpora that were given by the shared task organizers, no other preprocessing was used.

4.1. Monoses Pipeline

The training pipeline of Monoses consists of ten steps and produces a model for translation. For our purpose, we worked with the first eight steps of the pipeline and then extracted the needed information from the resulting phrase tables. The phrase-based translation models during the training are built with Moses (Koehn et al., 2007).

Firstly, Monoses preprocessed both corpora (for target and source language) – each corpus was tokenized, cleaned, truecased and split into training and development parts. In the second step, language models for both languages were trained. After that, phrase embeddings for extracted n-grams were trained with the help of the external tool Phrase2Vec (Artetxe et al., 2018c). The fourth step of the pipeline provided mapping of embeddings of phrases to cross-lingual space with the help of an external tool VecMap (Artetxe et al., 2018b). After that, the initial phrase table was induced for both directions (src to trg and trg to src). Next step built initial translation model for both directions. The seventh step is unsupervised tuning. This step was done using adapted MERT (Artetxe et al., 2019b). To run this step properly, the length initialization had to be chosen. That is because of different length of the input corpora. The last step we performed was the iterative refinement using back-translation. After the translation, the corpora were cleaned, and then aligned using FastAlign (Dyer et al., 2013). A Moses translation model was built from this aligned corpus and new phrase tables were produced. We used this step without tuning and we proceeded only one iteration of back-translation, because the corpora given for this task were too big.

As we discovered, the unsupervised tuning decreased the performance of the model for this particular task (see Table 5). Therefore, in our pipeline, we skipped the step number seven and used the model from the sixth step for further training. We also tried to run the pipeline on lemmatized corpora (preprocessed by Udpipe2 – see Section 3.1), but that decreased the performance as well (see Table 6).

4.2. Processing of Phrase Table

The phrase table created in the eighth step for translation from target to source was used to produce the results. Although the phrase table included all retrieved n-grams, we only considered the rows containing phrases from given lists of terms.

Each line of the phrase table contains a source phrase (in English for this task), a target phrase (in French) and several scores – inverse phrase translation probability, inverse lexical weighting, direct phrase translation probability, direct lexical weighting.

We needed only one score to sort the results according to their reliability, so we multiplied the direct and the inverse translation probability and used the product as the final score for the task. The result was then sorted according to this score and was submitted to the shared task (see example of results on train set: Table 2).

todo	todo	selection	sélection
desc	desc	slightly	légèrement
predecessor	prédécesseur	grade	grade
dramatic	dramatique	conversation	conversation
chapter	chapitre	tribe	tribu
literally	littéralement	mirror	miroir
iframe	iframe	choice	choix
fiction	fiction	formula	formule
propaganda	propagande	gang	gang
succession	succession	region	région
composition	composition	combination	combinaison
ritual	rituel	discussion	discussion
definition	définition	pilot	pilote
group	groupe	comparison	comparaison
compilation	compilation	coverage	couverture
survival	survie	source	source
birth	naissance	preparation	préparation
trackback	trackback	quiz	quiz
stats	stats	passage	passage
partnership	partenariat	resolution	résolution

Table 2: Top 40 translations on train set from second approach.

5. Combination of the Approaches

The main problem with the first approach is that it is not able to compare pairs with a different source term. We tried to overcome this problem by combining the results from the first and the second approach. Namely, we took pairs of terms acquired from the first approach and we arranged them in the following order: first the source terms with their nearest neighbour sorted by the scores obtained from the second approach, then the source terms with the second nearest neighbour again sorted by the scores from the second approach, etc. Table 3 contains top 40 translation for the train set. The results of this approach are summarized in Table 6.

6. Third Approach

Similar to the first approach, this method uses term embeddings to match corresponding term pairs based on their adjusted cosine similarity. It differs in the way we obtain the bilingual word embeddings and in the metric we use in the nearest neighbour search.

6.1. Corpus Preprocessing

We first matched the term occurrences in the training corpora and joined individual words composing a term with an underscore (e.g. phone_number, email_address). We then tokenized the corpora using

desc	desc	navigation	navigation
birth	naissance	group	groupe
tribe	tribu	formula	formule
composition	composition	population	population
difference	différence	conversation	conversation
generation	génération	region	région
combination	combinaison	existence	existence
neutral	neutre	gang	gang
choice	choix	selection	sélection
anti	anti	preparation	préparation
inhabitants	habitants	officially	officiellement
definition	définition	traditionally	traditionnellement
presence	présence	role	rôle
points	points	protection	protection
planet	planète	automatically	automatiquement
stock	stock	minutes	minutes
directly	directement	massage	massage
possession	possession	resolution	résolution
distinction	distinction	easily	facilement
residence	résidence	creation	création

Table 3: Top 40 translations on train set from combination of the approaches.

the Hugging Face pretrained tokenizers[1] to modify the input into the form each model expects it.

6.2. Multilingual Language Models

In contrast to the static embeddings used in the first approach, we experimented with contextualized embeddings from multilingual BERT (Devlin et al., 2018) and XLM (Conneau and Lample, 2019) model. The models we used have 12 and 16 layers, respectively, each of which encodes every subword into a vector of 756 and 1280 elements, respectively. We followed the method of Kvapilikova et al. (2020) to bring the XLM embeddings closer together by fine-tuning the model on a small portion of parallel sentences using the TLM objective (Conneau and Lample, 2019). According to the previous research, the parallel sentences used for fine-tuning do not have to match the language pair of interest so we experimented with English-German sentences from the News Commentary as well as English-French data provided for this task. The English-French parallel sentences were mined from the monolingual training data using the LASER sentence embeddings (Artetxe and Schwenk, 2019) where we retrieved the first 300,000 matching pairs. We also experimented with monolingual fine-tuning on the training corpora using the masked language model (MLM) (Devlin et al., 2018) objective.

The fine-tuning was performed in the XLM toolkit (`https://github.com/facebookresearch/XLM`) provided by the authors of the model with the Adam (Kingma and Ba, 2015) optimizer and the learning rate of 0.00005.

[1] `https://huggingface.co/docs/transformers/main_classes/tokenizer`

6.3. Term Embeddings

We took the embeddings from the 5th-to-last layer of the models as the mid-layers of the models carry the most multilingual information (Kvapilikova et al., 2020; Pires et al., 2019). Each word is composed of subwords and some terms have more than one word. We calculated the contextualized term embedding by averaging the embeddings of the subwords it contains. The embeddings are context-dependent. In order to get rid of this dependence, we took an average of the contextualized embeddings for one term over all the contexts from the training data set. This method is also referred to as the average anchor method (Schuster et al., 2019).

6.4. Term Retrieval

We used cosine similarity with Cross-modal Local Scaling (CSLS) (Conneau et al., 2018) to retrieve the term translation candidates. To compile the term dictionary, we keep only the closest candidate for each source term and sort the term pairs by their CSLC scores. The results are summarized in Table 7.

7. Evaluation

The evaluation of the task was done with the Mean Average Precision (MAP) metric. Our models produced a bilingual term pair list. The relevance of a term pair was determined by its presence in the gold dictionary ($D_{1,2}$).

Precision for *k* (see Formula 1) was computed as *k* divided by the size of the set of predicted term pairs from the top to the position where *k* relevant term pairs were retrieved (R_k).

Mean Average Precision (MAP) is then the sum over all *k* to the size of the golden dictionary (*m*) of precisions for *k* divided by *m* (see Formula 2).

$$P(R_k) = \frac{|R_k \cap D_{1,2}|}{|R_k|} \tag{1}$$

$$MAP = \frac{1}{m} \sum_{k=1}^{m} P(R_k) \tag{2}$$

7.1. Train Results

We present results for our three approaches on the train set (English-French language pair). The Table 4 presents results of the first approach. Results are divided according to the preprocessing used (UDPipe 1 or UDPipe 2) and to the number of computed nearest neighbours (for $k = 1, 2, 3, 5, 10$). The table shows the size of the terms dictionary, number of correct terms and Mean Average Precision. The best results were obtained for UDPipe 2 preprocessing and $k = 2$.

Overall we can conclude that this method is very precise when retrieving the nearest neighbour only. Including more neighbours increases the resulting dictionary size dramatically with only negligible effect on MAP. This effect is not that strong when using UDPipe

2 for preprocessing, presumably because it is more accurate and many of the second neighbours are translations already found in the other direction. As we have already mentioned, our theory is that the MAP does not raise significantly mainly, because we are unable to rank the translations correctly. Quite probably there are a lot of terms which have only 1 correct translation, however when using more neighbours, we include more translations for each of the terms, lowering the precision dramatically while raising the recall only marginally.

Method		Size	Correct terms	MAP
UDPipe 1	1nn	2504	2083	0.717
	2nn	4080	2178	**0.723**
	3nn	4452	2182	0.720
	5nn	6217	2218	0.712
	10nn	10113	2265	0.695
UDPipe 2	1nn	2225	1981	0.700
	2nn	3419	2131	**0.728**
	3nn	4459	2181	0.724
	5nn	6356	2237	0.713
	10nn	10398	2287	0.693
Gold dictionary		2519	2519	

Table 4: Approach 1 results on train set.

The scores for the second approach based on the phrased-based translation system are generally higher than for the first approach (see Table 5), but for the price of a bigger resulting dictionary. The best results were produced when skipping the tuning step and with no lemmatization during preprocessing.

Method	Size	MAP
Monoses – with tuning	6596	0.86
Monoses – no tuning	17087	**0.88**
Monoses – lemmatized, no tuning	31506	0.78
Gold dictionary	2519	

Table 5: Approach 2 results on train set.

As we can see from the results, this approach gets higher MAP score, however the sizes of the dictionaries are much bigger, so the model is not very precise. We assume it benefits strongly from the ability to rank produced translation pairs correctly, which allows it to get such a high MAP even with dictionaries that are big. It may be interesting to take only some of the highest scoring translations, we did not look into this though.

The results from the combination of the first two approaches are listed in the Table 6. When using UDPipe 1 for preprocessing, the best scores is obtained for $k = 1$. On the other hand, the best scores for UDPipe 2 preprocessing is acquired for $k = 10$. The overall best score is reached for $k = 10$ and UDPipe 2 for preprocessing, ordering the candidates according to the winning Monoses results.

Method			Size	MAP
Monoses with tuning	UDPipe 1	1nn	2504	**0.769**
		3nn	4452	0.762
		5nn	6217	0.760
		10nn	10113	0.750
	UDPipe 2	1nn	2225	0.766
		3nn	4459	0.833
		5nn	6356	0.838
		10nn	10398	**0.857**
Monoses without tuning	UDPipe 1	1nn	2504	**0.770**
		3nn	4452	0.761
		5nn	6217	0.759
		10nn	10113	0.750
	UDPipe 2	1nn	2225	0.772
		3nn	4459	0.842
		5nn	6356	0.857
		10nn	10398	**0.867**
Lemmatized monoses without tuning	UDPipe 1	1nn	2504	**0.764**
		3nn	4452	0.757
		5nn	6217	0.754
		10nn	10113	0.744
	UDPipe 2	1nn	2225	0.762
		3nn	4459	0.828
		5nn	6356	0.843
		10nn	10398	**0.851**

Table 6: Approach combination results on train set.

	Model	MAP
1	FastText + MUSE	0.859
2	bert-base-cased	0.783
3	xlm-mlm-100-1280	0.837
4	(3) + fine-tune MLM (en,fr)	0.871
5	(4) + fine-tune TLM (en-fr)	**0.897**
6	(3) + fine-tune TLM (en-fr)	0.880
7	(3) + fine-tune TLM (en-de)	0.881

Table 7: Approach 3 results on train set.

With this combination we have tried to leverage advantages of the two approaches – getting a better precision as the approach 1 and a better ranking as the approach 2. It mostly fulfilled our expectations, the best approach has only around $1\,\%$ lower MAP with a dictionary almost half the size.

We decided to submit three test runs according to these results, namely the term dictionaries from the first approach using UDPipe 2 preprocessing and $k = 2$, the second approach applied to raw corpora without tuning and their combination with $k = 10$.

The third approach was not finalized in time to be submitted to the official BUCC 2022 shared task on English-French term translation but we nevertheless include the results on the train set for completeness and comparison. Given the favorable results, we planned to use this approach for the German and Russian test sets, possibly in a future round of this task.

All scores in Table 7 were obtained using the CSLS metric for nearest neighbour search and dictionary creation. We compare contextualized multilingual embeddings from the 5th-to-last layer of the pretrained models with a baseline of static bilingual embeddings with 100 elements trained by FastText and aligned using MUSE with no supervision and see that the pretrained models do not reach the baseline but the XLM-100 model performs significantly better than the BERT-base model. Fine-tuning the XLM-100 model on the task-specific texts provided for training brings the results over the baseline, especially when using the quasi-parallel sentences retrieved by LASER. Interestingly, in agreement with the findings of (Kvapilikova et al., 2020), fine-tuning on completely unrelated parallel sentences (English-German) leads to an almost identical improvement.

8. Conclusion

We designed three approaches to bilingual term alignment. We searched for the nearest neighbours in the term embedding space created by a static FastText embedding model with post-hoc alignment (Approach 1) and pretrained multilingual language models (Approach 3). We also used an unsupervised phrase-based machine translation system created from the training data and searched its phrase tables for term pair candidates (Approach 2). The latter approach leads to similar results on the train set but only the Approach 1 and Approach 2 were finished in time to be submitted for the test run.

We learned that the pretrained multilingual model XLM-100 and its universal contextualized embeddings lead to a similar performance as static embeddings trained on the task-specific training corpus. However, the static embeddings have a significantly lower embedding size (300 in contrast to 1280 of the XLM-100 model) so the comparison is not straightforward. When fine-tuning the XLM model with task-specific data, we were able to push the precision higher from 0.837 to 0.897 (MAP on train set).

9. Acknowledgements

This work has received funding from the grant 19-26934X (NEUREM3) of the Czech Science Foundation, and support from the project "Grant Schemes at CU" (reg. no. CZ.02.2.69/0.0/0.0/19_073/0016935). This research was partially supported by SVV project number 260 575.

10. Bibliographical References

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

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

Artetxe, M., Labaka, G., and Agirre, E. (2018a). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia, July. Association for Computational Linguistics.

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

Artetxe, M., Labaka, G., and Agirre, E. (2018c). Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium, November. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (2019a). Bilingual lexicon induction through unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5002–5007, Florence, Italy, July. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (2019b). An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy, July. Association for Computational Linguistics.

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

Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. In H. Wallach, et al., editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2018). Word translation without parallel data. In *6th International Conference on Learning Representations (ICLR 2018)*.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv [e-Print archive]*, abs/1810.04805.

Dyer, C., Chahuneau, V., and Smith, N. A. (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 Tech-*

nologies, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.

El-Kishky, A., Chaudhary, V., Guzmán, F., and Koehn, P. (2020). CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 5960–5969, Online, November. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Kvapilikova, I., Artetxe, M., Labaka, G., Agirre, E., and Bojar, O. (2020). Unsupervised multilingual sentence embeddings for parallel corpus mining. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. In print.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July. Association for Computational Linguistics.

Ravi, S. and Knight, K. (2011). Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 12–21, Portland, Oregon, USA, June. Association for Computational Linguistics.

Schuster, T., Ram, O., Barzilay, R., and Globerson, A. (2019). Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1599–1613, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Shi, H., Zettlemoyer, L., and Wang, S. I. (2021). Bilingual lexicon induction via unsupervised bitext construction and word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 813–826, Online, August. Association for Computational Linguistics.

Straka, M. and Straková, J. (2017). Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August. Association for Computational Linguistics.

Straka, M. (2018). UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October. Association for Computational Linguistics.