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

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

Abstract

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

Keywords: BDI, BWE, Orthography, Transliteration

1. Introduction

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

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

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

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

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

*The authors contributed equally to this manuscript.

signal.

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

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

2. BUCC 2020 Shared Task

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

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

3. Approach

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

3.1. Bilingual Word Embeddings

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

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

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

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

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

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

3.2. Orthographic Similarity

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

3.3. Transliteration score

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

3.3.1. Transliteration model

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

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

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

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

3.3.2. Cleaning process

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

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

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

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

3.4. Dynamic Translation

BDI is often performed by returning the top-1 or top-5 most probable translations of a source word (Mikolov et al., 2013b). Since the dictionaries of the shared task contain

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

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

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

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

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

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

4. Experimental Setup

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

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

5. Results

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

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

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

	High			
	En-De	De-En	En-Ru	Ru-En
FTT cbow	38.17	46.37	33.52	46.78
Surface	4.31	3.41	7.38	14.64
Ensemble	40.59	49.56	38.33	54.12
Ensemble + Margin	39.76	49.90	36.23	54.71
Wiki	41.40	48.61	39.43	54.90
	Middle			
	En-De	De-En	En-Ru	Ru-En
FTT cbow	30.62	36.00	20.14	39.82
Surface	7.76	10.11	13.47	16.93
Ensemble	47.76	51.71	33.24	49.64
Ensemble + Margin	47.76	51.89	36.17	49.72
Wiki	49.18	53.66	43.55	56.53
	Low			
	En-De	De-En	En-Ru	Ru-En
FTT cbow	24.19	33.05	15.03	21.53
Surface	24.62	20.12	20.62	30.25
Ensemble	63.82	69.41	30.11	42.99
Ensemble + Margin	63.82	69.41	30.50	43.17
Wiki	65.14	73.10	51.72	57.01

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

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

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

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

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

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

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

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

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

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

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

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

6. Conclusion

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

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

Acknowledgements

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

Bibliographical References

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

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

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

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

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

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

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

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

Czarnowska, P., Ruder, S., Grave, E., Cotterell, R., and Copestake, A. (2019). Don't Forget the Long Tail! A Comprehensive Analysis of Morphological Generalization in Bilingual Lexicon Induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 974–983.

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

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

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

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

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

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

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

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

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

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

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