Bilingual Lexicon Induction from Comparable Corpora

Indian Institute of Technology BHU(Varanasi), India
Arvind Agrawal, Devadutta Dash
{arvind.agrawal.cse19, devadutta.dash.cd.cse19} @itbhu.ac.in

Abstract

The paper focuses on Bilingual Lexicon Induction (BLI) using comparable corpora, which is an integral framework of machine translation (MT). We have worked on language pairs such as English-German, German-English, English-French, French-English, English-Spanish, Spanish-English and Hindi-Bhojpuri. Our strategy comprises of : (i) A mapping approach based on projecting the monolingual word embeddings into the target space and then estimating the translations by calculating the CSLS similarity scores ; (ii) A perfect cognate approach based on exact match string similarity; (iii) A character level transformer model for low resource languages. We have combined the first 2 approaches for all the language pairs (except Hindi-Bhojpuri), and it gives us quite promising results (F1 score,precision and recall). To deal with out-of-vocabulary words for low resource language pairs,which are closely related,such as Hindi-Bhojpuri, we have implemented a transformer model, trained on the character-level embeddings and the Hindi-Bhojpuri cognate pairs, to generate the bilingual dictionary which gives us a respectable bleu-score and precision.

Keywords : Bilingual lexicon induction, Comparable corpora, Cognates, Word embeddings, Transformer model

1 Introduction

Bilingual Lexicon Induction (BLI) is the task of inducing word translations using monolingual corpus of the two languages. It is also used to evaluating the quality of the bilingual word embedding (BWE) models (Mikolov et al., 2013)(Vulić and Korhonen, 2016). In this paper we have worked on several languages including English, German, French, Spanish, Hindi and Bhojpuri and have used different techniques to induce good quality dictionaries for different language pairs.

Cross-lingual word embeddings have caught the attention of many researchers and several approaches have been tried to align them so as to map and generate words of different languages using unsupervised, semi-supervised and supervised methods along with different types of similarity scores to map bilingual word embeddings and generate bilingual word pairs with the help of mapped embeddings. These methods work fine with high or mid frequency words but fail to generate good quality bilingual dictionaries for the low frequency word pairs. We have modified the approach given by (vecmap)(Artetxe et al., 2018) and have combined it with a perfect cognates matching approach (perfect string matching) to predict the translations using certain hyper-parameters, which have been tuned according to the language pair and the frequency distribution of words. Our strategy particularly performs well for the mid and low frequency words giving quite promising results and given the limited computational resources that we currently have we get a pretty decent score on the high frequency words as well.

In this paper, we have also worked on a common problem encountered in low resource languages for which there is not enough corpora to generate good quality word embeddings. The vecmap (Artetxe et al., 2018) approach, therefore,is not able to generate good quality bilingual dictionaries due to the presence of out-of-vocabulary (OOV) words. To resolve this issue we have taken a closely related language pair(Hindi and Bhojpuri in this case) and tried to implement a character level transformer model that exploits the morphological information present in inter-character interactions.

In the rest of the paper we have described our approach to the task of generating Bilingual Dictionary,the hyper-parameters specific to each language pair and the future prospects we wish to fulfill. Then we present our results on the BUCC 2020 Dataset and the hindi-bhojpuri dataset(link) and finally present our conclusion.

2 Related Work

Bilingual Dictionary induction is the task of inducing word translations from monolingual corpora in different languages. This task is the basis for all the language translation models because the first thing we require is word to word translation for translating any language. It is also one of the main tasks used for evaluating the quality of BWE models (Mikolov et al., 2013) (Vulić and Korhonen, 2016). The role of BLI in tasks such as translating out-of-vocabulary words in Machine Translation is also quite significant (Huck et al., 2019).

Bilingual Lexicon induction requires corpora in some form or the other to train the model, parallel corpora was initially widely used for training but training without parallel corpora has been also there for a significant time starting with the influential works of (Rapp, 1995). The models by (Fung and Yee, 1998); (Rapp, 1995);(Schafer and Yarowsky, 2002);(Koehn and Knight, 2002);(Haghighi et al., 2008);(Irvine and Callison-Burch, 2013) used statistical similarity between the languages to learn small dictionaries. Recent studies and usage of word embeddings have shown the improvement in statistical decipherment (Dou et al., 2015). The methods involving cross lingual word embeddings basically focuses on aligning the word embedding spaces of both the languages. Several methods which use these embeddings learn mappings from the source to the target space and then using nearest neighbour kind of approach to find translations (Mikolov et al., 2013); (Zou et al., 2013); (Faruqui and Dyer, 2014); (Ammar et al., 2016). (Mikolov et al., 2013) showed how to train the model using small seed lexicon. Also (Conneau et al., 2017) and (Artetxe et al., 2018) are able to learn BWEs without any seed dictionaries using a self-learning method that keeps on improving from a weak solution.

But this method usually worked with high frequency word pairs and failed with low frequency word pairs due to the poor vector representations of such words as shown in some papers (Riley and Gildea, 2018); (Czarnowska et al., 2019); (Braune et al., 2018). (Braune et al., 2018) showed improved results on rare and domain specific words by using character n-gram and levenshtein similarity. But then also we were not able to obtain significant result.

BUCC shared task 2020(Rapp et al., 2020) aimed at generating bilungual dictionary induction from comparable corpora so as to find some reasonable solution for the low frequency word pair problem. (Laville et al., 2020) introduced an approach wherein they introduced the perfect cognate which we have used in the low frequency word pairs because most of the low frequency words are words like proper nouns which have the same translation in both the languages. (Severini et al., 2020) used sim delete algorithm for same script languages and transliteration model for languages such as en-ru which have different scripts. Our model focuses on low resource languages wherein it uses an approach similar to (Laville et al., 2020) for foreign language pairs and a character level transformer model for low resource languages which are very closely related and exhibit a good amount of phonetic and orthographic similarities.

3 Dataset

We have generated the cross lingual mapped embeddings by using the semi-supervised method given by vecmap (Artetxe et al., 2018) and as a training dictionary, we have used the muse dictionaries given by facebook research (Conneau et al., 2017). We have taken the word embeddings from BUCC 2020(Rapp et al., 2020), where they have provided pre-trained fasttext skipgram embeddings trained on WACKY or Wikipedia corpora.

Language	en	fr	de	es
Corpus	Wacky	Wacky	Wacky	Wikipedia

Table 1: Corpus used for the languages

We were not able to train our model using the supervised setting on vecmap because of computational limitations. Therefore, we have used the semi-supervised setting which trains the model on a seed lexicon of 20,000 words which we have taken from facebook research (Conneau et al., 2017). For validating the results, we have used the training dictionary provided by the BUCC 2020(Rapp et al., 2020), as the testing dictionaries have not been disclosed by them.

Lang.-	unique src words			Total Translations			ratio		
pairs	high	mid	low	high	mid	low	high	mid	low
en-de	2000	2000	2000	5852	3608	2296	2.92	1.804	1.15
de-en	2000	2000	2000	4295	3346	2454	2.14	1.673	1.227
en-fr	2000	2000	2000	4385	3198	2240	2.19	1.59	1.12
fr-en	2000	2000	2000	3276	2834	2334	1.63	1.41	1.17
en-es	2000	2000	2000	4770	3065	2235	2.39	1.53	1.18
es-en	2000	2000	2000	3508	2817	2301	1.75	1.41	1.15

Table 2: Test-Data Description along with **ratio** of target words per source word

For Hindi-Bhojpuri we have taken the pretrained fasttext embeddings given by facebook and using them we have generated the character level embeddings (Jha et al., 2018). The training and validation dictionaries have been taken from (Jha et al., 2018).

Training Word Pairs	3201
Validation Word Pairs	799

Table 3: Training and Validation Data for Hindi-Bhojpuri transformer model

4 Approach

To induce bilingual dictionaries from comparable corpora, the basic idea is to learn an efficient transfer matrix between the source and the target words that preserves translation pairs proximity of a seed lexicon. After the mapping step, a similarity measure is used to rank the translation candidates for a particular source word. The approach given by (vecmap) (Artetxe et al., 2018) has 2 major drawbacks : (i) it generates only one translation for a particular word in the source language and we know that one particular source word can have multiple translations in the target language depending on the context (ii) it does not give a proper strategy to deal with the low frequency and out of vocabulary(OOV) words in an efficient manner. We have dealt with both of these problems in our paper. In the following subsections we have described the strategies used.

4.1 BWEs,Vecmap and Cognates-Matching

For the foreign language pairs : english-german, german-english, english-french, french-english, english-spanish and spanish-english we have tried an approach wherein we have taken the pre-trained embeddings available on BUCC 2020(Rapp et al., 2020) which were trained using the following hyperparameters : minCount: 30; dim: 300; ws (context window): 7; epochs: 10; neg (number of negatives sampled): 10. We have used the vecmap(Artetxe et al., 2018) to project by pairs every source embeddings space in its corresponding target space. We have used the semi-supervised approach given by vecmap and as a seed lexicon we use the muse dictionary given by facebook research(Conneau et al., 2017). The reason for using the semi-supervised method is : significant portion of the muse dictionary contained OOV translations and the effective size of dictionary available to map the embeddings was reduced and whenever we tried supervised method we were faced with cuda out of memory error due to computational limitations. Once the embeddings were projected into a common space, we compared every source word in the validation set to every word in the target vocabulary using a similarity measure. We used the CSLS(Cross-Domain Similarity Local Scaling) Conneau et al. (2017), which is based on the cosine similarity and 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 :

$$\boldsymbol{CSLS(}x_s,\ y_t\boldsymbol{)}\ =\ \boldsymbol{2cos(}x_s,\ y_t\boldsymbol{)}\ \text{-}\ \boldsymbol{knn(}x_s\boldsymbol{)}\ \text{-}\ \boldsymbol{knn(}y_t\boldsymbol{)} \tag{1}$$

To generate multiple translations for a particular source word we set a ***threshold limit(k)*** and ***max-translations(n)*** and we only consider those words in the target vocabulary for which the similarity score is above the threshold limit and if the number of such words come out to be greater than the max-translations, then we select the first n words (here max-translations = n) having the highest similarity scores. If the situation arises that no word in the target vocabulary has the similarity score above the threshold, in that case we have considered the perfect cognate of the source word as the translation. The reason for doing this is : when we analysed the low frequency word pairs, we realised that for most of the source and target words pairs, the translations are graphically identical and many were even perfect cognates of each other. Therefore we added that if for a particular word, the similarity scores of all the words in the target vocabulary are below the threshold, then the translation for that word is most likely a perfect cognate that does not exist in the target vocabulary. This is because the hyper-parameters used to train the fasttext embeddings takes the minimum count of the occurrence of a word as 30 and the frequency of occurrence of the word pair translations which are perfect cognates of each other is really less.

We have followed a similar approach for the OOV(out of vocabulary) words where for each particular source word present in OOV words we have printed the perfect cognate of the word as the translation. For the high-frequency word pairs, we have implemented one additional strategy which increases the F1 score of the bilingual lexicon induced by a considerable amount. After executing the above mentioned approach, we check whether,for every source word in the validation set, a perfect cognate exists in the target vocabulary. If such a perfect cognate pair exists that has not been taken into account by the CSLS similarity approach, we add that translation pair to the bilingual lexicon.

	en-de			de-en			en-fr		
Frequency	high	mid	low	high	mid	low	high	mid	low
Threshold	0.1	0.75	0.75	0.1	0.68	0.7	0.1	0.75	0.7
max-translations	2	2	1	2	2	1	2	2	1
	fr-en			en-es			es-en		
Frequency	high	mid	low	high	mid	low	high	mid	low
Threshold	0.1	0.72	0.7	0.1	0.7	0.72	0.1	0.72	0.72
max-translations	2	2	1	2	2	1	2	2	1

Table 4: Hyperparameters for the language pairs

Algorithm 1 Vecmap and Perfect cognates

```
for each source and target language corpora do
    Generate word embeddings using fasttext
    Stack the word embeddings of words of the source and target corpora to form the matrices X and Z respectively of the dimension(embedding size x number of words in vocabulary)
    Use vecmap to learn the linear transformation matrices Wx and Wz so that the mapped embeddings XWx and ZWz are in the same cross-lingual space
end
for each source word in the validation set do
    Use CSLS to calculate the similarity score of every possible target word
    Sort the similarity scores from highest to lowest and choose a similarity threshold (k)
    Take the target words(T) having similarity score $\geq$ threshold as possible translations for the source word and fix the max-translations(n) for a particular source word
    L = length(T)
    if If $L \geq n$ then
        Take the top n target words as possible translations for the source word
    else
        Take the T as the list of possible translations for the source word
    end
    if $L = 0$ or the particular source word is an OOV word then
        Take the perfect cognate of the source word as a possible translation
    end
    if the source word is of high frequency and a perfect cognate(P) of the source word exists in the target vocab but doesn't exist in T then
        Include P as a possible translation of the source word
    end
end
```

This approach gives quite promising results for the mid and low frequency word pairs and for the high frequency words our results are comparable to that of those of the teams which participated in BUCC 2020 shared task.

4.2 Transformer Model

For low resource languages such as Hindi and Bhojpuri, the vecmap approach failed due to the following three reasons : (i)Since the corpora available is very limited, the quality of the fasttext embeddings generated was not up to the mark (ii)Moreover, due to the limited corpora available the problem of OOV(out of vocabulary) words arises which leads to very low F1 score (iii) The pre-trained fasttext embeddings contained most of the english words which were just changed into hindi with the same pronunciation and not actually translated.

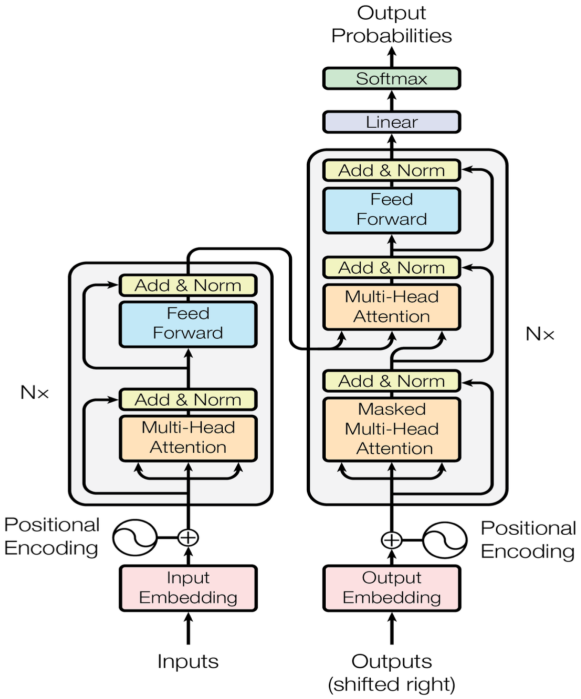

Figure 1: The Transformer architecture as described in and adapted from (Vaswani et al., 2017)

We have tried to implement a Seq2seq NMT-transformer model to address the issue encountered in vecmap. The language pair used by us is Hindi and Bhojpuri. The fundamental concept behind our approach is the fact that Bhojpuri and Hindi are closely related languages, and therefore have a good amount of vocabulary overlap. Moreover the vocabulary of Hindi and Bhojpuri exhibit orthographic and phonetic similarities. Both these languages have common ancestors, and both of them are written in the Devanagari script. The traditional Phrase Based Machine Translation (PBMT) approach (Chiang, 2005) or NMT(Bahdanau et al., 2014) for Bhojpuri becomes infeasible in this context as it requires a massive parallel corpora, and we don't have access to such corpora. Therefore to take care of this problem and create a machine translation system with limited corpora, we tried to implement a character level transformer model that exploits the morphological information present in inter-character interactions. We used the character level embeddings of Hindi, which have trained using the fasttext embeddings of Hindi. A character-level embedding is obtained by averaging over all word vectors of words in which the character occurs, weighted by the number of times it occurs in each word (Jha et al., 2018). Using the Open-NMT framework, we trained a transformer model on Hindi Bhojpuri cognate pairs. Instead of performing machine translation using words, the transformer model works on the characters as the fundamental units. We then evaluated the model against a validation set and then calculated the precision and bleu score.

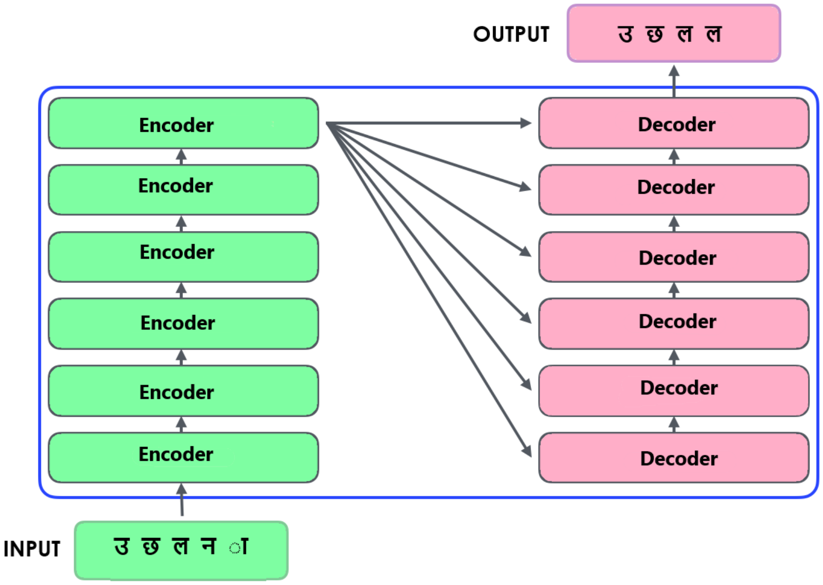

Figure 2: Hindi-Bhojpuri Lexicon induction using Transformer architecture

As shown in the above diagram instead of words as input to the transformer encoder, we use the characters of the source word in Hindi. The corresponding output word is obtained character by character in Bhojpuri at the decoder .

Optimisation		Batching		Model	
model dtype	fp32	encoder type	transformer	queue size	10000
optimiser	adam	decoder type	transformer	bucket size	32768
learning rate	2	position encoding	true	world size	4
warm up steps	8000	encoder layers	6	gpu ranks	[0,1,2,3]
decay method	noam	decoder layers	6	batch type	tokens
adam beta2	0.998	heads	8	batch size	512
max grad norm	0	rnn size	512	valid batch size	8
label smoothing	0.1	transformer ff	2048	max generator batches	2
param init	0	dropout$_{s}teps$	0	accumulator count	4
param init glorot	true	dropout	0.1	accumulator steps	0
normalization	tokens	attention dropout	0.1		

Table 5: Hyper-parameters tuned for the Transformer Model On OpenNMT Framework

5 Results

Similar to the official evaluation metric of the BUCC shared task 2020 we calculated Precision Recall and F1 score for all the language pairs, the procedure followed is as given in (Rapp et al., 2020). We compared the results for different sets of hyper-parameters by trying out different values of threshold and max-translations and the final hyper-parameters were as mentioned in table-4.

Results by Frequency										
La ng.	Team Name	HIGH			MID			LOW		
		P	R	F1	P	R	F1	P	R	F1
en-de	LS2N	51.6	49.0	50.3	52.6	53.7	53.2	65.2	68.6	66.9
	ours	32.14	54.56	40.45	54.38	39.44	45.72	73.55	64.07	68.48
de-en	LS2N	63.7	48.1	54.8	63.0	59.0	60.9	73.3	72.2	72.8
	ours	42.34	52.01	46.68	52.91	42.41	47.08	75.05	61.17	67.40
en-fr	LS2N	66.2	55.2	60.2	67.6	59.9	63.5	78.5	74.4	76.4
	ours	36.69	49.81	42.26	58.83	44.37	50.59	79.65	71.12	75.14
fr-en	LS2N	65.6	54.6	59.6	64.3	49.1	55.7	62.0	29.4	39.8
	ours	41.79	67.43	51.60	58.52	48.59	53.09	75.95	65.08	70.09
en-es	LS2N	61.7	57.6	59.6	56.8	63.3	59.9	67.1	77.8	72.1
	ours	35.85	52.87	42.72	53.70	46.46	49.81	72.90	65.23	68.85
es-en	LS2N	74.9	61.9	67.8	72.8	66.4	69.4	78.0	77.2	77.6
	ours	40.43	57.92	47.62	58.19	49.20	53.32	74	64.32	68.82

Table 6: Precision, Recall and F1 score comparison between LS2N as published by BUCC and our approach

The test set on which BUCC has published the results by using the approach of LS2N has not been made public by the them. Therefore we have used the training dictionary provided by the BUCC 2020(Rapp et al., 2020) to validate the results. Moreover, we have trained our model on a seed lexicon which is approximately one-fifth of that used by LS2N. But the results our quite promising. The fact that several of the low frequency pairs are words like proper nouns which have a perfect cognate existing in the other language as well worked in our favour.

For hindi-bhojpuri character level transformer model we got the following results.

Accuracy	25.82
Bleu-Score	82.35

Table 7: Accuracy and bleu-score for Hindi-Bhojpuri transformer model

5.1 Analysis

When we studied the training dictionaries provided by the BUCC organisers, we noticed the presence of multiple words not belonging to the correct language. For example in the English to French training lexicon words like god,northwest,phoenix, gov and many others are present on the French side which do not actually belong in the French language and we think manual supervision is required to get rid of them. We even find erroneous word pairs with none of the words belong to either of the source and target languages. We also find many proper nouns in the training lexicon and most of them are graphically identical words and therefore our perfect-cognates approach works particularly well in case of low frequency word pairs (For eg. Gabrial, Johnstown on English to French test set(BUCC training set)).

Results by Frequency										
La	Method	HIGH			MID			LOW		
ng.	Used	P	R	F1	P	R	F1	P	R	F1
en-	vecmap	43.32	27.80	33.87	45.96	22.70	30.39	31.86	23.56	27.09
de	vecmap with P.C.	32.14	54.56	40.45	54.38	39.44	45.72	73.55	64.07	68.48
de-	vecmap	41.12	33.32	36.81	43.10	28.93	34.62	40.28	24.57	30.52
en	vecmap with P.C.	42.34	52.01	46.68	52.91	42.41	47.08	75.05	61.17	67.40
en-	vecmap	43.26	32.20	36.91	44.72	28.71	34.96	32.90	20.18	25.01
fr	vecmap with P.C.	36.69	49.81	42.26	58.83	44.37	50.59	79.65	71.12	75.14
fr-	vecmap	42.95	39.80	41.31	44.44	28.62	34.81	33.55	20.01	25.07
en	vecmap with P.C.	41.79	67.43	51.60	58.52	48.59	53.09	75.95	65.08	70.09
en-	vecmap	39.32	37.47	38.47	48.16	36.17	41.31	44.45	31.87	37.12
es	vecmap with P.C.	35.85	52.87	42.72	53.70	46.46	49.81	72.90	65.23	68.85
es-	vecmap	44.12	41.05	42.53	52.19	38.45	44.27	45.65	31.46	37.25
en	vecmap with P.C.	40.43	57.92	47.62	58.19	49.20	53.32	74	64.32	68.82

Table 8: Analysis of the model and with and without Perfect Cognates(P.C.)

Moreover many source words have identical pairs on the target side even though they have proper translations. While the vecmap approach is unable to take care of these discrepancies, we have tried to take into account most of the word pairs by the perfect-cognates approach. As it can be observed from the table above, applying the perfect cognates approach along with vecmap brings about a considerable increase in F1 score for all the language pairs .

The results usually have a trend wherein the F1score is increasing from high to low frequency word pairs. This may be explained by the fact that low frequency word pairs are usually proper nouns or are graphically identical and because of this there are more chances of perfect cognates to exist. Also because perfect cognates are usually 1 translation per source word, the method performs well for low frequency word pairs. For the mid and low frequency words,we got the best results for max-translations 1 and 2 and similarity score threshold varying from 0.65 to 0.75 and for the high frequency words we got the best results for max-translations 2 and similarity score threshold varying from 0.05 to 0.2. While varying the hyper-parameters, we noticed that as we increase the number of translations in the predicted file, the precision decreases and the recall increases while the exact opposite occurs when we decrease the number of translations. The number of translations increased on decreasing the threshold or increasing the max-translations so we had to meet somewhere in between where the F1-score was highest.

For the transformer model, due to computational and implementation limitations, we could not perform such an extensive hyperparameter search. Nevertheless, we take care of the basic parameter settings by setting batch size to 512 (i.e., the largest such size avoiding BLEU score degradations and out-of-memory errors), and number of training epochs as 5000. The other hyperparameters that we have tuned for the model are given in the approach section 4.2 .

6 Conclusion and Future Prospects

In this paper we have presented our approach for Bilingual Lexicon Induction using comparable Corpora. We used pre-trained fasttext embeddings provided by the BUCC shared task 2020 organizers. We also used the graphical similarity between word pairs to improve our results based on a perfect cognate strategy. Overall our method was found to be efficient than the standard vecmap approach or perfect cognate approach alone. Moreover with the Hindi-Bhojpuri language pair we were not able to apply the BWE approach because of the limitation of corpus and thus resulting in bad quality BWE. However we implemented a character level transformer model that exploits the morphological information present in inter-character interactions between the 2 languages and were able to obtain promising results.

Due to computational inefficiencies we had to use the pretrained fasttext embeddings which were provided by the BUCC shared task organisers. Therefore as future work, we would like to implement five things : (i)The validation set used by us is different from that of LS2N. The test set on which LS2N validated their results has not been released by the BUCC yet. If it becomes available on the future, we would like to work on that. (ii) We would like to concatenate the fasttext skipgram and cbow embeddings to generate 600 dimensional word vectors and then use those embeddings for the bilingual lexicon induction task. (iii) To improve the F1 score for the low frequency and the mid frequency words, we would like to tune the hyperparameters of the fasttext embeddings to also include those words whose min count (frequency of occurrence of that word in the corpora) is less. (iv) In the unsupervised setting, vecmap(Artetxe et al., 2018) has generated the crosslingual mapped embeddings by using a self learning method, we would like to use adversarial learning to project the source embeddings into the corresponding target space. (v) Due to non-availability of resources and our inability to coordinate with other organisations we could not apply the transformer model on other closely related low resource language pairs which we would like to keep as a future work.

References

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

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv preprint arXiv:1805.06297*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Fabienne Braune, Viktor Hangya, Tobias Eder, and Alexander Fraser. 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, 2018.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*, pages 263–270, 2005.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

Paula Czarnowska, Sebastian Ruder, Edouard Grave, Ryan Cotterell, and Ann Copestake. Don't forget the long tail! a comprehensive analysis of morphological generalization in bilingual lexicon induction. *arXiv preprint arXiv:1909.02855*, 2019.

Qing Dou, Ashish Vaswani, Kevin Knight, and Chris Dyer. Unifying bayesian inference and vector space models for improved decipherment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 836–845, 2015.

Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, 2014.

Pascale Fung and Lo Yuen Yee. An ir approach for translating new words from nonparallel, comparable texts. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 414–420, 1998.

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

Matthias Huck, Viktor Hangya, and Alexander Fraser. Better oov translation with bilingual terminology mining. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5809–5815, 2019.

Ann Irvine and Chris Callison-Burch. Supervised bilingual lexicon induction with multiple monolingual signals. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 518–523, 2013.

Saurav Jha, Akhilesh Sudhakar, and Anil Kumar Singh. Learning cross-lingual phonological and orthagraphic adaptations: a case study in improving neural machine translation between low-resource languages. *arXiv preprint arXiv:1811.08816*, 2018.

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

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

Reinhard Rapp. Identifying word translations in non-parallel texts. *arXiv preprint cmp-lg/9505037*, 1995.

Reinhard Rapp, Pierre Zweigenbaum, and Serge Sharoff. Overview of the fourth bucc shared task: Bilingual dictionary induction from comparable corpora. In *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 6–13, 2020.

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

Charles Schafer and David Yarowsky. Inducing translation lexicons via diverse similarity measures and bridge languages. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

Silvia Severini, Viktor Hangya, Alexander Fraser, and Hinrich Schütze. Lmu bilingual dictionary induction system with word surface similarity scores for bucc 2020. In *Proceedings of the 13th Workshop on Building and Using Comparable Corpora*, pages 49–55, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Ivan Vulić and Anna Korhonen. On the role of seed lexicons in learning bilingual word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 247–257, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1024. URL `https://www.aclweb.org/anthology/P16-1024`.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, 2013.