Cross Lingual Transfer Learning for POS tagging

Ninad Khargonkar, Shreesh Ladha
{nkhargonkar, sladha} at umass dot edu

December 23, 2017

Abstract

Cross lingual embeddings have been gaining a lot of traction in the natural language processing community in recent years. We build upon this success and use them for performing POS tagging in a semi-supervised way. We use a pre-trained tagger in a high resource language to transfer its information to a low resource language by learning a transformation between the word embeddings. This has a lot of potential benefits for low resource languages like Hindi, for which annotated data is not readily available. We additionally use the tags as extra features in auxiliary tasks and obtain significant improvement in performance.

1 Introduction

Part of speech (POS) tagging is one of the fundamental tasks in the field of Natural Language Processing and it is often crucial to the performance of other high-level related tasks. It involves assigning a correct POS tag such as noun, verb or adjective to each word of a given sentence. This is usually modelled as a sequence labelling task using Markov models where the labels are POS tags. In spite of being spoken by more than a billion people, POS taggers for languages of Indian subcontinent are not being actively developed vis-a-vis other high resource languages such as English and French. This is partly due to the fact that there is not a lot of high quality annotated data available for training statistical models for these low-resource languages.

To overcome this barrier, our project is on exploring the possibility of transferring information from high resource languages such as English to improve the performance of POS-Taggers for languages with low resources, in our case, Hindi. To accomplish this task, we only require a tagger built on the source language and a parallel corpus or a bilingual lexicon between the source and target language pair. Prediction of POS tags for the target language is thus only semi-supervised and requires no information about the true POS tags for the source language.

To get a better sense of the quality of POS tags obtained with this method, we provide this information to other auxiliary tasks which may benefit from part of speech tag

information. In our experiments, we obtained a significant improvement in accuracies with models using these 'noisy' POS tags. This suggests that the method is useful and can be used in concatenation with some other task to improve performance. Henceforth, we'll be referring specific languages for better clarity, with Hindi being the source and English the target language respectively. To our knowledge, this is the first experiment of its kind (at least for Hindi).

2 Related Work

The task of part of speech tagging in English is a long studied problem in linguistics and language processing. However for languages like Hindi, it has only caught attention recently with a few groups working on it. Recent work on POS taggers for Hindi has only been limited to rule-based and probabilistic techniques. In the papers [1], [2] and [3], the authors use Hidden Markov Models (HMM) and Conditional Random Fields (CRF) to train their taggers. The HMM based tagger in [1] used simple word counts in the corpus to assign the transition and emission probabilities and Precision and Recall were used as evaluation metrics. In the work of [2], in addition to the base HMM model they also used a longest suffix matching stemmer for pre-processing the data before using it for training and it showed an improvement in performance, without using any other linguistic resource from the language.

A lot of recent work has focused on learning cross-lingual embeddings for a variety of NLP tasks. The aim of cross lingual learning is to learn a shared embedding space between words in all languages. This is motivated by the final goal of learning a single language-agnostic word embedding. Such techniques are now even leading to a way of translating languages without any parallel text. A few basic techniques have been discussed below.

In a method, proposed by **Mikolov et al**, 2013 [4], the author learns a transformation matrix to map a source language to a target language by reducing the least square error between the two and using a bilingual lexicon. The mono-lingual embedding of the languages is assumed to be known beforehand. Further work by **Xing et al** [5] and **Artetxe et al** [6] introduce some constraints on the learning process of the linear transformation like normalizing the vectors and making the transform matrix orthogonal in order to preserve some monolingual relations between words.

Most methods rely on sentence aligned parallel corpora for learning the transformation. **Guo et al, 2015**, [7], design a count matrix using parallel corpora of the two languages and comparing the context of two aligned sentences within the corpora. It then obtains an embedding of a target language by taking a weighted average of embedding vectors in the source language. **Faruqui and Dyer** [8], propose to use canonical correlation analysis (CCA) to project words from two languages into a shared embedding space.

There have been some methods that try to learn these embeddings without the use

of parallel text or lexicon. **Barone** [9], proposed such a method by analyzing the monolingual distribution of words. He maps word vectors from a source language to semantically compatible word vectors on a target language using an adversarial autoencoder. This is based on the assumption that different natural languages share similar semantic structures. We chose to use Mikolov's (along with its few derivatives) and Guo's method for our task. This was because they form the basis of all other techniques and require less data and time to implement.

3 Data

3.1 Univeral Dependencies

We use a POS-tagged data set for Hindi sourced from the Universal Dependencies project (UD). UD is a framework for cross-linguistically consistent grammatical annotation and an open community effort with over 200 contributors producing more than 100 tree-banks in over 60 languages. The data set for Hindi language already has in built train and test splits. The test split is used as ground truth (for reporting accuracy and other measures) in methods which assume no Hindi-POS information during training i.e the pseudo-unsupervised methods. A bried summary for the Hindi section of the data set is included below:

- The training set contains 281059 tokens spread over 13305 sentences
- The test set contains 35217 tokens over 1660 sentences.
- It also contains tag information on a both coarse and fine level. The coarse level pos tags are shown in the Table 1.
- Some words have more than one possible POS tag assigned to them based on the context in the sentence they are present in. The exact distribution for the counts is described in Table 2.

3.2 HindEnCorp

HindEnCorp is a English-Hindi parallel text corpus collected from several sources like EMILLE dataset, Wikipedia texts and TED talks captions. This parallel data set is used for transforming Hindi word vectors to their English counterparts (as a crude word level translation). It contains 273886 sentences (3.9 million Hindi and 3.8 million English tokens). The corpora has been collected from web sources and pre-processed primarily for the training of statistical machine translation systems. This dataset is parallel alingned at the sentence level.

ADJ	adjective	**ADP**	adposition
ADV	adverb	**AUX**	auxiliary
CCONJ	coordinating conjunction	**DET**	determiner
INTJ	interjection	**NOUN**	noun
NUM	numeral	**PART**	particle
PRON	pronoun	**PROPN**	proper noun
PUNCT	punctuation	**SCONJ**	subordinating conjunction
SYM	symbol	**VERB**	verb
X	other		

Table 1: Coarse level tags in UD

Number of tags per word (n)	No. of words with "n" tags	Percentage
1	15773	87.14
2	2032	11.23
3	256	1.41
4	30	0.17
5	8	0.04
6	1	0.01

Table 2: Different possible POS tags per word in UDP

3.3 Hindi Monolingual Text Corpus

This is a Hindi text Corpus obtained from Government Of India's **TDIL** initiative. It has documents from 8 different domains and each document for a domain consists of 1001 sentences. We used this data set for a domain classification task since the documents also contain ground truth part of speech tag information. The domains are as follows: agriculture, entertainment, literature, philosophy, politics, economy, religion and sports.

3.4 Hindi-English lexicon

A **dictionary** mapping between English and Hindi words containing 136110 word pairs, created by the Center for Indian Language Technology at IIT-Bombay. The top 5000 most common word pairs were extracted from this dictionary and these pairs were used in the training process of learning the linear transformation.

4 Method

The overall pipeline has been described in Figure 1. Given a word in Hindi we obtain its closest counterpart in English using a transformation learnt using one of the Cross Lingual Learning techniques (described in a later section). We repeat this process for all the words to get a sentence solely in English. This is now fed to a pre-trained English POS tagger. POS tags thus obtained could now be used as it is, or fed to an auxiliary task for better learning. We will refer to this method elsewhere in the document as **UPOS** (unsupervised part of speech).

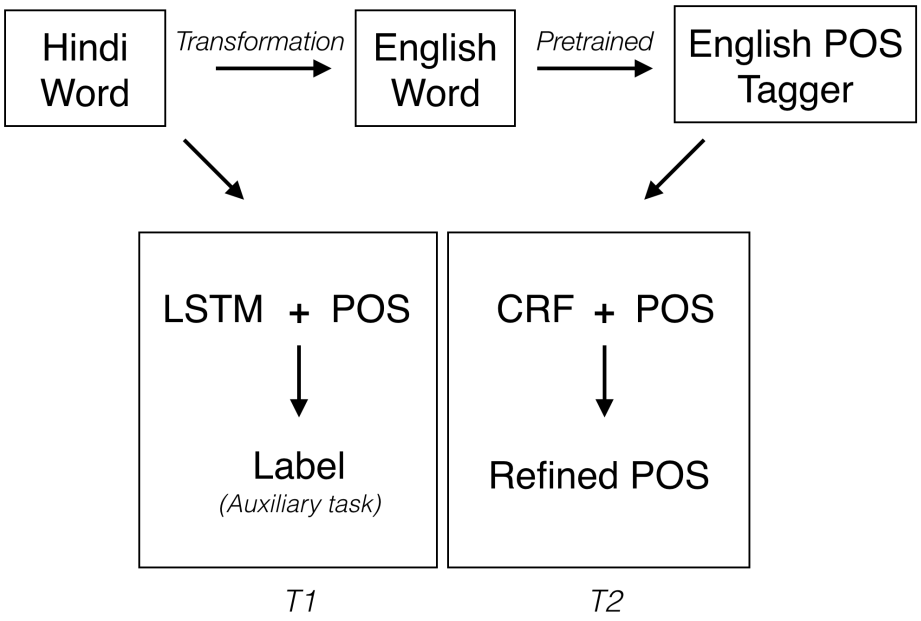


Figure 1: Method Pipeline

A basic pipeline has been outlined in Figure 1. For T1, POS refers to the POS tags obtained using the method described above and LSTM refers to sentence representation obtained using that architecture. The information from those two can be concatenated to predict a label. The second task 'T2' is the task of refining a POS tagger that you already have by feeding these 'noisy' POS tags during training as features to the CRF training algorithm. This in a sense provides some prior information to the model. Details have been discussed later in this section.

The initial stage of the process looks very similar to translation but the caveat is that in general translation two constraints are relaxed which we require for our algorithm to work:

- The number of words in the Hindi sentence might not be equal to the number of words in the translated version

- Even if the above condition holds, there is no guarantee that the i_{th} word in the Hindi sentence would correspond to the i_{th} in the translated English sentence

These are required so that we have a one-to-one mapping between words in the two languages and thus the POS tags can be assigned to the Hindi words using this mapping. We are actually doing something similar to a word by word translation. The difference is that cross lingual techniques require much less data. Also even a translation of a single word might have two words in the translated language which breaks our algorithm. In the following subsections we will go through the different approaches for obtaining the transformation.

4.1 Baseline

We used CRF as the baseline algorithm for POS tagging. The training set that was used for this model is the UD training set that was described above and the same test set provided along with it was used for testing. We trained our model both using the CRF code that we used in our homework as well as off-the-shelf toolkit CRFsuite. We used the same set of features as in our homework i.e a bias and word_postag term for this version. The HW code was run for 10 iterations, while the CRFsuite toolkit was run for 100 iterations since it was much faster.

4.2 Cross Lingual Transfer Learning

We tried to learn transformations to map a word in a source language to a target language using monolingual word embeddings. The transformations are learned in a 300 dimensional word vector embedding space and we use the pre-trained fastText vectors released by Facebook.

4.2.1 Linear transformation

Mikolov et al.[4] propose a method for obtaining a word level translation between two languages using a linear transformation for the monolingual word embeddings in both the languages. This method assumes the availability of pre-trained monolingual embeddings for both the source and target language. According to the authors, the motivation for learning such a linear transformation is the observation that geometric relations between word vectors are similar across languages.

As a consequence this gives rise to the suggestion that the two spaces might be related through a linear transformation. The process of learning the linear projection matrix is achieved by taking the top 5000 translation pairs from a bilingual dictionary between the source and target language. Once we have the set of translations pairs (and hence their embeddings), the matrix W can be found out as a solution to the following:

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

$$\text{or} \tag{1}$$

$$\min_W \sum_{i=1}^{n} ||XW - Z||_F^2$$

Here x_i is the embedding vector of the translation pair in the source language (Hindi) and z_i is vector for the word in target language (English) and similarly X and Z are the matrix representations for the entire training set. The solution can be found out by either using stochastic gradient descent or trying to obtain the closed form solution directly, $(X^T X)^{-1} X^T Z$.

We also tried out some modifications in the original problem formulation. Artetxe et al. [6] further introduce some constraints which generalize the work of previous papers and improve upon the original method by Mikolov et al. [4].

They enforce the matrix W to be orthogonal ($WW^T = I$) along with the normalizing the vectors as seen in [5]. Their motivation was to preserve the dot products while the learning the mapping and getting a cosine similarity measure since the vectors are now normalized. Additionally another constraint of dimension wise mean centering was enforced on the word embedding matrices X and Z for the training set to make the expected similarity between two randomly chosen words to go to zero. This is done by multiplying the X and Z matrices by their corresponding dimension wise centering matrices:

$$\arg\min_W ||C_m X W - C_m Z||^2 \tag{2}$$

4.2.2 Count Matrix

Guo et al.[7] propose a projection method that relies on word alignments. They count the number of times each word in the source language is aligned with each word in the target language in a word aligned parallel corpus and store these counts in a count matrix $\mathcal{A}$.

In order to project a word w_i from its source representation $v(w_i^S)$ to its representation in the target embedding space $v(w_i)^T$ in the target embedding space, they simply take the average of the embeddings of its translations $v(w_j)^T$ weighted by their counts with the source word.

$$v(w_i)^T = \sum_{i,j \in \mathcal{A}} \frac{c_{i,j}}{\sum_j c_{i,j}} \cdot v(w_j)^T \tag{3}$$

where $c_{i,j}$ is the number of times the i^{th} source word has been aligned to the j^{th} target word. They set the projected vector of an out of vocabulary source word as the average

of the projected vectors of source words that are similar to it in edit distance.

4.3 Domain Classification

This was used to test the quality of the POS tags that we obtained using UPOS. Specifically, we first index the 4000 most common words in our dataset and mark the rest as unknown. We then clip the sentences having more than 20 words in their sentence (The average was 13 per sentence). Now for a given sentence, we first obtain its 64 dimensional dense embedding and then pass it through an LSTM layer with dropout probability as 0.2. This is further passed through a single fully connected layer with eight outputs. For the second experiment, we use the POS information obtained using UPOS and create a bag of words representation. This is simply concatenated to the feature obtained after passing the sentence through the LSTM, which is now fed to the fully connected layer just as before.

4.4 Baseline with Prior

This is the second experiment we tried to assess the quality of UPOS tags. We feed these POS tags as features to our baseline CRF model.

5 Results

We used the English POS tagger provided by nltk as the pre-trained POS tagger. The possible tagset is the 'universal' tagset which contains: ('.', 'ADJ', 'ADP', 'ADV', 'CONJ', DET', 'NOUN', 'NUM', 'PART', 'PRON', 'VERB', 'X'). The UDP data for Hindi that we used for testing the accuracy of our model had some minor differences in the labelling for the tags. We handled such cases manually in our scripts. For example CONJ (conjunctions) were split into coordinating and subordinating conjunctions. There was an additional tag for Proper Nouns which was merged into parent tage of Nouns. Similarly, AUX (auxiliary verbs) were mapped to Verbs before computing the accuracy.

5.1 Baseline

We compared two types of accuracies, token level and sentence level. Token level compares if a word's correct tag was predicted and sentence level compares if the entire sentence's correct tagset was predicted. We obtained high token level accuracies of 95% and 97% on the training dataset with the HW and CRFsuite code respectively. Similarly high accuracies were obtained on the test set as well, being close to 92% and 93% for the two different methods. The detailed tag-report having precision-recall scores has been described in Table 3.

We believe that the test accuracies were high primarily because the test and training data are from similar sources (news data). Secondly majority of Hindi words have just one possible tag (details in the table described in Dataset section), hence there isn't a

lot of ambiguity regarding the choice. That being said, the sentence level accuracies were still relatively on the lower side. On the training set, the sentence level accuracy obtained was **57.71%** while on the test set it was **38.80%**. Detailed precision-recall scores can be seen from Table 3.

Label	Precision	Recall	F1-score	Support
X	0	0	0	4
PART	0.986	0.967	0.976	722
CCONJ	0.981	0.985	0.983	682
SCONJ	0.98	0.994	0.987	682
ADJ	0.936	0.872	0.903	2144
ADP	0.984	0.993	0.989	7380
ADV	0.805	0.651	0.72	292
VERB	0.948	0.935	0.941	3302
DET	0.941	0.95	0.945	699
INTJ	0	0	0	0
NOUN	0.891	0.938	0.914	7928
PRON	0.985	0.961	0.973	1473
PROPN	0.874	0.842	0.858	4214
NUM	0.97	0.849	0.905	715
PUNCT	1	1	1	2367
AUX	0.953	0.962	0.958	2613
Avg	0.939	0.939	0.939	

Table 3: Precision-Recall scores for CRF based method

5.2 Cross Lingual Transfer Learning

5.2.1 Linear transformation method

We were unable to learn a good transformation using Mikolov et al's [4] algorithm. This might have happened because the algorithm tries to learns a linear transformation while the embedding vectors for our pair might be related by a non-linear transformation. We tried using a 2 layer neural network as well to mitigate this non-linearity issue but were unable to get it working with that as well. This suggests that this method might work only for languages that share the same morphology such as English, Spanish, German etc. unlike Hindi which is quite different.

For a better picture we've visualized the differences between the vectors in 2-dimensions in Fig. 1, 2 & 3 (dimensions are reduced by using Principle Component Analysis). Fig.1 shows vectors for 50 Hindi vectors reduced in 2 dimensions and Fig.2 shows the vectors

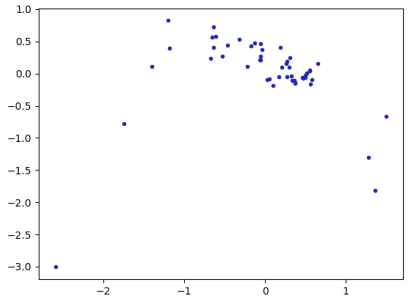


Figure 2: Hindi vectors

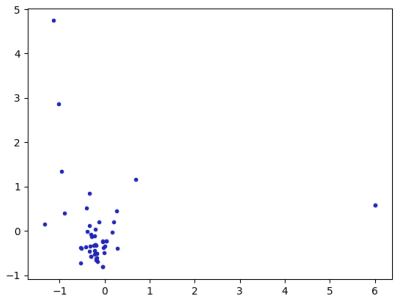


Figure 3: English vectors

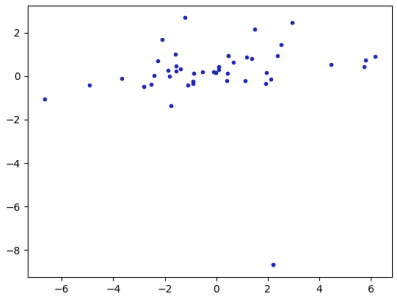


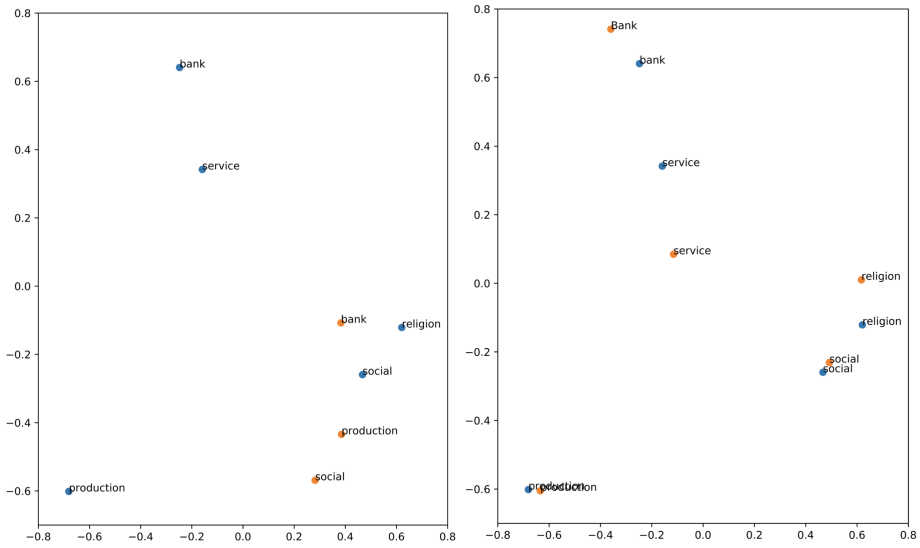
Figure 4: Hindi transformed to English

for the corresponding English words. Vectors are the embeddings obtained from fastText. Fig.3 shows the transformed vectors (i.e Vectors obtained after transforming the Hindi vectors using the learned linear transformation). Ideally these should be similar to the true English vectors in Fig.2 but we can see this is not the case.

5.2.2 Count Matrix

The transformation for Hindi words that we obtained using this method [7] gave much better results qualitatively (see Figure 5). We obtained an accuracy of **66%** on the test set (UD). Note that this entire process in completely semi-supervised in the way that it requires no information for the Hindi POS tags, it just requires a tagger for English. The detailed precision-recall scores for various tags is shown in Table 5 and an illustration of the visualization can be seen in Figure 5.

One observation was that this method faced problems on a few selected POS tags like determiners and conjunctions. This may be due to the fact that such tags may have slightly different context in Hindi and English unlike proper nouns which are somewhat universal in nature. For example in Hindi, the relatively common word 'और' has an ambiguous tag (it can other than a conjunction depending on the context) but translates to 'and' in English and thus gets mapped to a conjunction.



(a) Using Linear Transform (b) Using Count-Matrix

Figure 5: Common words with their ground truth (blue) and transformed projections (orange). Count matrix projection method is able to map two semantically similar words in Hindi and English close to each other while the linear transform projection is unable to capture this relationship.

Method	Accuracy
CRF	**93.9**
Count Matrix	66.6

Table 4: Token level Accuracy on UD Treebank

5.3 Domain Classification

We split the dataset into proportion of 75-25. The inital 75%, 6006 sentences, was used for training while the remaining 25% 2002 sentences for testing. The accuracies obtained have been tabulated in Table 6. You can see that there is a significant improvement of 11.8% in accuracies when UPOS information is provided to the model.

5.4 Baseline with Prior

For this experiment we deliberately worked on a smaller subset of 5000 sentences from the original UD dataset. This was done since using UPOS prior information for this task makes sense when the original data, is not enough to generalize the model. This theory was supported by our experiments. With the entire data, we didn't obtain any improvement in accuracies using the UPOS tags. On a smaller dataset we obtained an improvement of 0.6% on the token level and 1.58% on the sentence level accuracies.

Label	Precision	Recall	F1-score
X	0	0	0
PART	1.0	1.3e-11	2.7e-11
CONJ	0.931	0.396	0.556
ADJ	0.567	0.508	0.536
ADP	0.604	0.607	0.605
ADV	0.079	0.352	0.13
VERB	0.702	0.760	0.730
DET	0.116	0.597	0.195
NOUN	0.854	0.751	0.799
PRON	0.805	0.690	0.743
NUM	0.901	0.507	0.649
PUNCT	0.991	0.694	0.817
AUX	0.953	0.962	0.958

Table 5: Precision-Recall scores for Count Matrix based method [7]

Method	Accuracy
LSTM	72.25
LSTM + Count Matrix	**84.05**

Table 6: Accuracy on Hindi Monolingual Text Corpus

Method	Token Acc.	Sentence Acc.
CRF	92.2	30.53
CRF + Count Matrix	**92.8**	**32.11**

Table 7: Accuracy on subset of UD Treebank

6 Conclusion

In our project we worked on obtaining POS tags for a particular language using semi-supervised algorithms. To achieve this, we used pre-trained English POS tagger along with different cross lingual learning techniques to obtain closest Hindi-English pairs. The accuracies obtained in this fashion were reasonable but their real impact was observed on combining them in a different task. This motivates one to use this technique in other auxiliary tasks as weak supervision where large amounts of annotated POS data is not available. With regards to Cross Lingual Embedding techniques, the linear transformation methods didn't work as well as the count matrix method for POS

tagging probably because the latter benefitted from parallel data and relied less on the morphological structure of the two languages.

Further work in this area might include obtaining the English words with a pre-trained translator (with some manual tricks to handle multiword scenarios). This experiment would help us better understand the impact of Cross Lingual embedding techniques against general translation. We couldn't try it out since there wasn't any free Hindi-English translator available (Google has a limit on the conversions one can do). Another thing we couldn't try out (due to a lack of dataset) is testing these POS tags for an auxiliary task for which ground POS tags are also known (eg. named entity recognition). If this technique would have yielded accuracies close to that obtained using the grouth truth POS tags, we could've essentially shown that POS features obtained using our weak-supervision method are just as good for the task. This would further reduce dependency on obtaining completely annotated POS data for low resource languages which is a common problem.

References

[1] Nisheeth Joshi, Hemant Darbari, and Iti Mathur. Hmm based pos tagger for hindi. 2013.

[2] Manish Shrivastava and Pushpak Bhattacharyya. Hindi pos tagger using naive stemming : Harnessing morphological information without extensive linguistic knowledge. 2008.

[3] Siva Reddy and Serge Sharoff. Hindi pos tagger.

[4] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013.

[5] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *HLT-NAACL*, pages 1006–1011, 2015.

[6] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *EMNLP*, pages 2289–2294, 2016.

[7] Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. Cross-lingual dependency parsing based on distributed representations. In *ACL*, 2015.

[8] Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *EACL*, 2014.

[9] Antonio Valerio Miceli Barone. Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. *CoRR*, abs/1608.02996, 2016.

[10] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[11] Daniel Zeman, Martin Potthast, Milan Straka, Martin Popel, and Dozat et el. CoNLL 2017 shared task system outputs, 2017. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

[12] Ondřej Bojar, Pavel Straňák, Daniel Zeman, Gaurav Jain, and Om Prakesh Damani. English-hindi parallel corpus, 2010. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.