Cross-Lingual word embeddings for syntactically dissimilar languages by POS tags based word-alignment

Yash Patel

June 30 , 2020

M.Tech. Project Report

Abstract

In this data driven world of natural language processing, where abundant textual data is available for several languages, most languages have minimal written system or literature available. Neural network approaches to a range of NLP problems have also been aided by initialization with word embeddings[1]. In low-resource scenario we cannot expect to have parallel corpora available in order to learn the word embeddings through supervised learning. This encourages us to look for unsupervised approaches that transfer semantic information from a source language(generally high resource language i.e. english) to a target language[2]. The unsupervised transfer of semantic information between languages is a challenging task which becomes even more challenging when the languages are syntactically dissimilar i.e. grammatically dissimilar.

Firstly this project deals with obtaining monolingual embedding such that there exists no syntactical /grammatical information in the embeddings via Dependency parse trees. On top of these pre-trained embeddings, it aims to generate high quality embeddings for low-resource language via Cross-Lingual Word Embedding model and subsequently generate a bilinugual dictionary in the form of word-alignment task. The results have been reported on word-alignment task with and without using dependency parse tree to generate monolingual embeddings on news dataset[3] for various languages.

1 INTRODUCTION

The dependency parse tree is a structure that captures syntactic structure of a language. Apart from representing syntactic structure, dependency trees have been found suitable for solving problems that require semantic relations between elements of sentences[4]. These problems include the discovery of inference rules for question answering (Lin and Pantel 2001) and paraphrase identification (Szpektor, Tanev, Dagan, and Coppola 2004). There are various methods to generate a set of patterns that can identify instances of relations aka pattern models[5]. These models are evaluated in order of coverage of the corpus.

There exists many languages in the world which do not have proper literature available. Neural network approaches to a range of NLP problems have been aided by initialization with word embeddings trained on large amounts of unannotated text[1]. In case of low-resource languages we do not have much unannotated text. This problem was solved using the concept of cross lingual word embeddings. Mikolov et al. (2013b) first noticed that continuous word embedding spaces exhibit similar structures across languages. They proposed that using linear transformation one can map source language embeddings(generally resource rich language) to target language embeddings.

Both supervised and unsupervised techniques aim to find a linear transformation that maps word embeddings from source language to target language[2]. In low resource scenario the unsupervised techniques are the only possible way to transfer semantic information between languages. This is already a challenging task which becomes even more challenging when the languages are syntactically dissimilar i.e. grammatically dissimilar. In this challenging scenario many techniques fails to give promising results.

This project aims to tackle this dissimilarity with

the help of dependency tree, in order to generate quality monolingual word embeddings. These pre-trained embeddings are then used to generate high quality embeddings for low resource language using Cross-Lingual word embeddings model[6] with the knowledge of POS tags trained for word-alignment task. This report presents results using a modified version of the architecture mentioned in [6] for generating the cross lingual word embeddings and reports precision values for word-alignment task [6].

2 RELATED WORKS

A lot of investigation has been done in the field of cross-lingual word embeddings. Many methods are supervised i.e. require parallel corpora or comparable corpora to connect the languages (Klementiev et al., 2012[7];Hermann and Blunsom, 2013[8]; Chandar A P et al.[9], 2014; Coulmance et al., 2015[10]; Wang et al., 2016), or use bilingual dictionaries (Mikolov et al., 2013b; Gouws and Søgaard, 2015[11]; Duong et al., 2016[12]; Ammar et al., 2016)[13], while other are unsupervised (Conneau et al. 2018[6]; Artetxe et al. 2018[14]; T. Wada et al. 2019)[15].

The application of Dependency parse tree have been shown for various NLP problems i.e. Question Answering[16][17], Paraphrase Identification[18][19]. But usage of dependency parsing tree for cross-lingual word embeddings still remains unexplored.

3 METHODOLOGY

This report draws on work in two general areas, which we briefly describe in this section.

3.1 Syntax-Free Monolingual word embeddings using Dependency parse tree

Dependency parse tree: A Dependency Tree is a structure that can be defined as a directed graph, with |V| nodes (vertices), corresponding to the words, and |A| arcs, corresponding to the syntactic dependencies between them. Figure 1 shows an example of a dependency parse tree. A dependency grammar approach abstracts away from word-order information, representing only the information that is necessary for the parse. Apart from representing syntactic structure, dependency trees are regarded as a suitable basis for semantic patterns acquisition as they abstract away from the surface structure to represent relations between the elements of the sentences[4].

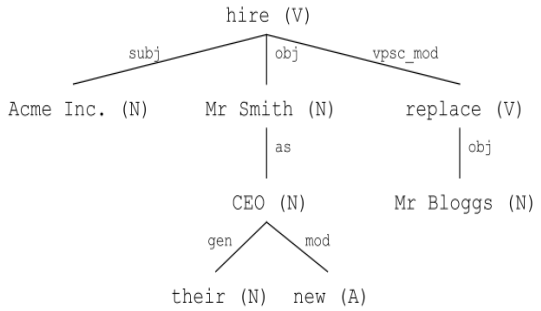


Figure 1: Dependency parse tree for "Acme Inc. hired Mr Smith as their new CEO, replacing Mr Bloggs."[5]

Pattern Models: [5] mentions different pattern models to obtain the semantic relationship between words. Various pattern models are listed below:

Predicate-Argument (SVO): A simple approach, used by Yangarber (2003), Stevenson and Greenwood (2005), is to use subject-verb-object tuples from the dependency parse as extraction patterns as shown in Figure 2.

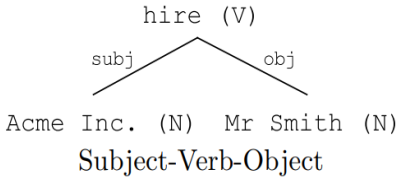
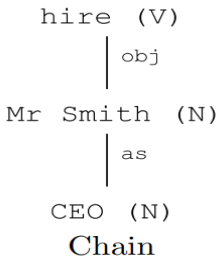


Figure 2: Example of pattern obtained from SVO pattern model"[5]

Chain: A pattern is defined as the direct path between a verb node and any of its descendants, passing through zero or more intermediate nodes (Sudo, Sekine, and Grishman 2001) as shown in Figure 3.



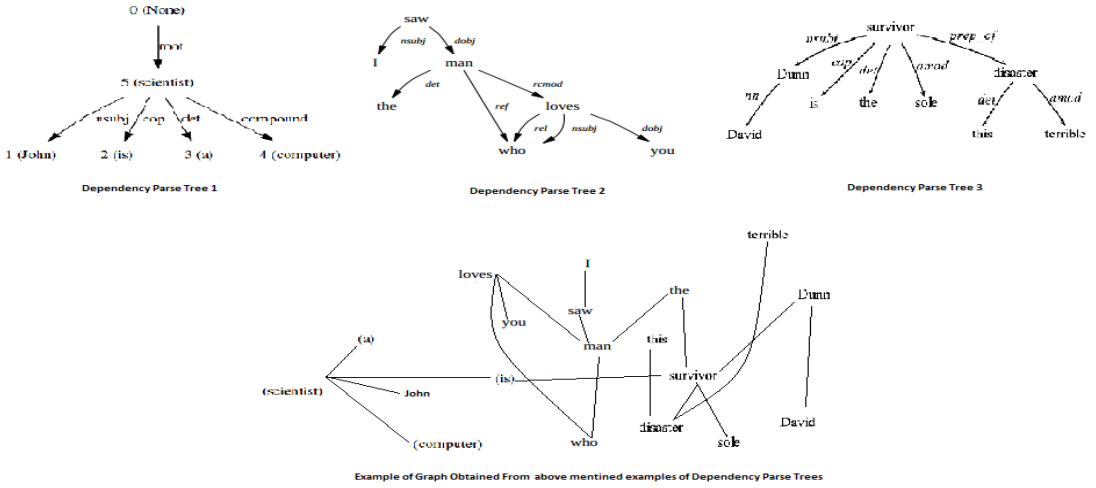


Figure 6: Example of Graph obtained from above mentioned technique

Figure 3: Example of pattern obtained from Chain pattern model"[5]

Linked Chain: The linked chain model, introduced by Greenwood et al. (2005), represents extraction patterns as a pair of chains which share the same verb as their root but do not share any direct descendants as shown in Figure 4.

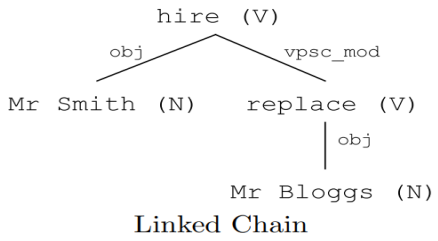


Figure 4: Example of pattern obtained from Linked Chain pattern model"[5]

Subtree: The final pattern model to be considered is the subtree model, introduced by Sudo et al. (2003). In this model any subtree of a dependency tree can be used as an extraction pattern, where a subtree is defined as any connected subset (possibly improper) of nodes in the tree. An example is shown in Figure 5.

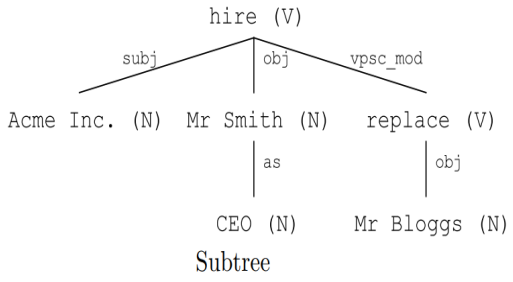


Figure 5: Example of pattern obtained from Subtree pattern model"[5]

It can be easily concluded that these techniques mentioned above work by studying the links that join the words in the sentence. The relationship is obtained between the words using the links having direct or indirect contact with each other.
In this project we propose to obtain the semantic relationship between the words using graph based techniques i.e. node2vec[20] and GraphSAGE[21] on dependency parse tree.

Obtain monolingual embeddings using Graph based techniques:

In this project we obtained a graph from the dependency parse tree, by connecting a word to every other word that is linked to it directly in a dependency parse tree as shown in Figure 6.

Learning useful representations from highly structured objects such as graphs is useful for a variety of machine learning applications. We used graph-based techniques mentioned below to obtain the word embeddings in the form of node embeddings:

node2vec[20] is a flexible biased random walk algorithm which includes both Breadth-First Sampling(BFS) and Depth-First Sampling(DFS) which is helpful in exploiting both structural as well as the homophilic nature of the graph. The node2vec framework learns low-dimensional representations for nodes in a graph by optimizing a neighborhood preserving objective which helps it to generate corpus from a graph data structure. The objective is flexible, and the algorithm accomodates for various definitions of network neighborhoods by simulating biased random walks. node2vec is a modification of DeepWalk[22] with the small difference in random walks. It has parameters p and q. Figure 7 shows an example of the random walk procedure in node2vec

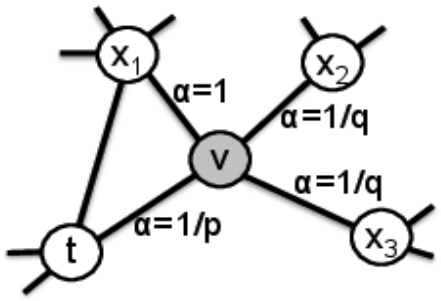


Figure 7: Example of the random walk procedure in node2vec[20].

Parameter p defines how probable it is that the random walk would return to the previous node. If p is low, then random walk samples nodes which are closer to itself which exploits the neighborhood nature. While in constrast, if p is high, random walk samples nodes which avoids 2-hop redundancy in sampling, i.e., it tries not to sample the node from where it had taken the last step. The parameter p controls discovery of the microscopic view around the node.

Parameter q defines how probable it is that the random walk would discover the undiscovered part of the graph.The parameter q controls the discovery of the larger neighborhood. If q $<$ 1, nodes are more likely to be sampled in DFS fashion, i.e., nodes which are further away from the node t are most likely to be explored while in contrast, if q $>$ 1, nodes are more likely to be sampled in BFS fashion, i.e., nodes which are closer to node t are more likely to be explored.

After transitioning to node v from t(shown in Figure 7), the return hyperparameter, p and the inout hyperparameter, q control the probability of a walk staying inward revisiting nodes (t), staying close to the preceding nodes (x1), or moving outward farther away (x2, x3). It infers communities and complex dependencies present in the dependency parse trees.

GraphSAGE[21], a general inductive framework that leverages node feature information (e.g., text attributes) to efficiently generate node embeddings for previously unseen data as shown in Algorithm 1. Instead of training individual embeddings for each node, it learns a function that generates embeddings by sampling and aggregating features from a node's local neighborhood.

Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K; weight matrices $\mathbf{W}^k, \forall k \in \{1, ..., K\}$; non-linearity σ; differentiable aggregator functions $\text{AGGREGATE}_k, \forall k \in \{1, ..., K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$

Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

$\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$;
for $k = 1...K$ **do**
  for $v \in \mathcal{V}$ **do**
    $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$;
    $\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$
  end
  $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$
end
$\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

Algorithm 1: GraphSage algorithm from [20].

3.2 Cross-lingual word embeddings

3.2.1 Word embeddings without parallel data

The word embeddings of source and target language trained independently using the technique mentioned in Section 3.1 is assumed. This is similar to that proposed by Mikolov et al. (2013b) which learns the linear transformation from source to target language such that:

$$W^\star = \underset{W \in M_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_{\mathrm{F}}$$

where d is the dimension of the embeddings, $M_d(R)$ is the space of d $\times$ d matrices of real numbers, and X and Y are two aligned matrices of size d x n containing the embeddings of the words in the parallel vocabulary.

[6] proposes an unsupervised model which learns the mapping matrix W using a domain-adversarial approach. Its model is trained to discriminate be-

tween elements randomly sampled from $WX = \{Wx_1, ..., Wx_n\}$ and Y. W is trained to prevent the discriminator from making accurate predictions. As a result, this is a two-player game, where the discriminator aims at maximizing its ability to identify the origin of an embedding, and W aims at preventing the discriminator from doing so by making WX and Y as similar as possible. This approach is in line with the work of Ganin et al. (2016), who proposed a technique which learns a latent representation invariant to the input domain. In our case, a domain is represented by a language (source or target).

Discriminator objective: We refer to the discriminator parameters as θ_D. We consider the probability $P_{\theta_D}(\text{source} = 1|z)$ that a vector z is the mapping of a source embedding (as opposed to a target embedding) according to the discriminator. The discriminator loss can be written as:,

$$\mathcal{L}_D(\theta_D|W) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}\left(\text{source} = 1\middle|Wx_i\right) - \frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}\left(\text{source} = 0\middle|y_i\right)$$

Mapping objective: In the unsupervised setting, W is now trained so that the discriminator is unable to accurately predict the embedding origin,

$$\mathcal{L}_W(W|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}\left(\text{source} = 0\middle|Wx_i\right) - \frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}\left(\text{source} = 1\middle|y_i\right).$$

So, a neural network model with stochastic gradient updates is trained to minimize $\mathcal{L}_D$ and $\mathcal{L}_W$.

In this project we used an upgraded variant of the model mentioned[6]. This model aims to generate the cross lingual word embeddings by learning the transformation matrix from source language to target language using a domain-adversarial approach. As can be seen from the loss function, the word that is the correct image(in target language)corresponding to the word in source language, is mapped by aligning along the whole target language vocabulary. Our model uses the POS(part of speech) tags obtained from the dependency parse tree and allow the model to search for its corresponding image in target language. Unlike the model mentioned above which maps the image of word in source language by aligning along the whole target language vocabulary, our model maps the image by aligning among the words with similar POS tags.

We propose an unsupervised model which learns the mapping matrix W using the domain adversarial approach taking POS tags into account. It is a model which contains the input in the form of a tuple i.e. (x_1, t_1), (x_2, t_2), (x_3, t_3).....,(x_n, t_n) where x_i is the word in source language, t_i is the POS tag corresponding to x_i and T_{dis} is the set of distinct POS tags corresponding to source and target vocabulary. It is a model trained to discriminate between elements sampled from W (X_T) = W (x_1, t_1), W (x_2, t_2), W (x_3, t_3)....., W (x_n, t_n) and Y_T. Unlike sampling the elements randomly from W X_T and Y_T, our technique samples the elements randomly from a subset obtained from W X_T and Y_T having similar POS tags.

4 DATASETS

4.1 Dependency parse tree

Universal Dependencies (UD) is a project that seeks to develop cross-linguistically consistent treebank annotation, by both providing annotation guidelines and releasing freely available treebanks. Apart from this, treebank artificial dataset is generated on the News Dataset using the StanfordNLP 0.2.0[23] which is a Python natural language analysis package. StanfordNLP features full neural network pipeline for robust text analytics, including tokenization, multi-word token (MWT) expansion, lemmatization, part-of-speech (POS) and morphological features tagging and dependency parsing.

4.2 Monolingual Corpus

Experiments are performed on News Dataset for English, French and Hindi languages. Basis is a list of about 32,000 news sources in more than 120 languages provided by ABYZ News Links.[3]

4.3 Bilingual Dictitonaries

[6] created 110 large-scale ground-truth bilingual dictionaries using an internal translation tool. The dictionaries handle the polysemy of words adequately. They have provided a train and test split of 5000 and 1500 unique source words, as well as a larger set of up to 100k pairs. Bilingual dictionaries to evaluate the quality of Cross-Lingual word embeddings are the ones published by [6].

5 EVALUATION METRICS

Word-alignment Task: As a measurement of quality of word embeddings, we used cross-domain similarity local scaling (CSLS), which is mentioned in [6]. CSLS is used as an improved version of the nearest neighbor method, as nearest neighbors are by nature

Techniques / Language Pairs		English-French	English-Hindi	English-German	English-Finnish	English-Italian	English-Japanese	English-Dutch	English-Russian
CLWE [6]	P@1	81.47	0.23	**73.77**	40.22	75.01	**0.001**	77.05	48.15
	P@5	**90.79**	0.81	86.65	64.32	86.79	**0.007**	**88.84**	**74.23**
	P@10	**92.80**	0.89	**89.47**	**71.23**	89.12	**0.007**	**91.32**	**79.52**
CLWE with POS tags	P@1	**82.12**	**0.32**	72.93	**42.71**	**76.19**	0.0	**77.42**	**48.16**
	P@5	89.23	**1.08**	**87.50**	**64.48**	**88.13**	0.005	86.417230	71.63
	P@10	91.45	**1.72**	89.23	70.64	**90.72**	0.005	89.478856	77.38

Table 1: Word-alignment average precision P@1, P@5 and P@10 on Cross-lingual Word Embedding(CLWE) [6] and Cross-lingual Word Embedding model with POS tags
(Note: These results are generated on fasttext embeddings published by [24], the pre-trained embedding generated from technique mentioned in Section 3.1 is not considered)

asymmetric: y being a K-NN of x does not imply that x is a K-NN of y. In high-dimensional spaces, this leads to a phenomenon called hubness. CSLS can mitigate the hubness problem in high-dimensional spaces, and can generally improve matching accuracy. It takes into account the mean similarity of a source language embedding x to its K nearest neighbors in a target language:

$$rT(x) = \frac{1}{K} \sum_{y \in \mathcal{N}_T(x)} \cos(x, y)$$

where *cos* is the cosine similarity and $\mathcal{N}_T(x)$ denotes the K closest target embeddings to x. rT(y) is defined in a similar way for any target language embedding y. CSLS(x, y) is then calculated as follows:

$$\text{CSLS}(x, y) = 2\cos(x, y) - rT(x) - rS(y)$$

6 Experiments

Experiments involve models from two different areas in NLP. Initially, the word embeddings using Monolingual data are independently generated using node2vec model[22] which generates a corpus from the graph obtained by collating the dependency parse trees. The corpus obtained from random walks of node2vec is passed to fasttext model[24] to generate monolingual word embeddings. We also tried the GraphSAGE model[21] to generate the embeddings using Inductive framework on the graph generated by collating the dependency parse trees. GraphSAGE is a technique that has expertise in Inductive representation learning on large graphs, the graph obtained by collating the dependency parse tree is large enough. Now these embeddings are used as input to the cross lingual word embedding model using POS tags from Section 3.2.1 to align the embeddings from source to target language, such that words with similar context are mapped closer when plotted. These embeddings are obtained by training the above mentioned model using knowledge of POS tags obtained from dependency parse trees. The evaluation is done on bilingual dictionaries published by [6] and the metric used to measure the quality of the embeddings is same as mentioned in [6] i.e. CSLS(Cross-Domain Similarity Local Scaling).

The output of the Cross-Lingual Word Embedding model using knowledge of POS tags would be the word embeddings for both source and target language in the same space. The target language embeddings should contain the transferred semantic information from the source language as firstly, we have made the corpora grammar/syntax free using our technique mentioned in section 3.1 and secondly making the word-alignment process more precise by instead of mapping the image of word in source language by aligning along the whole target language vocabulary, our model maps the image by aligning among the words with similar POS tags. These techniques should ensure high quality word embeddings as compared to embeddings obtained from using normal monolingual corpus with different grammars/syntactical rules and not using POS tags for word-alingment.

7 Results

Table 1 shows the comparison of performance on Word-alignment task as average precision score P@1, P@5 and P@10 between Cross-Lingual Word Embeddings obtained from method used in [6] and the technique mentioned in section 3.2.1 on fasttext embeddings published by [24] for English, French, Hindi, Dutch, German, Italian, Finnish, Japanese and Russian languages. The results are reported using 80K

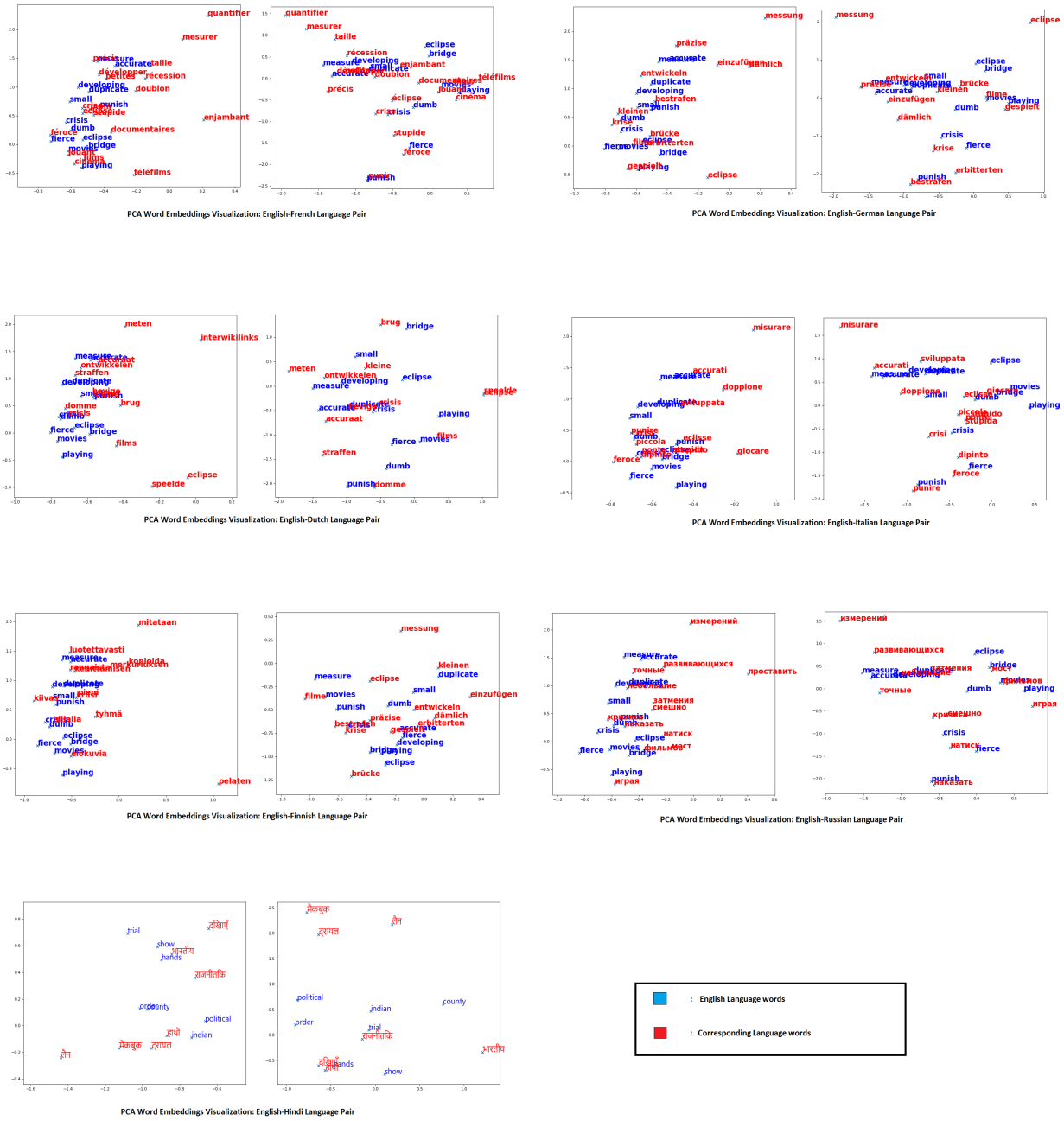


Figure 8: PCA Word Embeddings visualisation for various language pairs. Among the plot pairs, left image corresponds to the PCA visualisation on word embeddings obtained from Cross-lingual Word Embedding(CLWE) model [6] and the right image corresponds to the PCA visualisation of word embeddings obtained from Cross-lingual Word Embedding model with POS tags
(Note: These results are generated on fasttext embeddings published by [24], the pre-trained embedding generated from technique mentioned in Section 3.1 is not considered)

sized vocabularies for language pairs where English is considered as the source language in all cases and French, Hindi, Dutch, German, Italian, Finnish, Japanese and Russian as the target languages. The results are reported in the form of $P@1$, $P@5$ and $P@10$ CSLS Similarity Scores, where precision $P@m$ is "how often the correct translation of a source word x_i is included in the m extracted target words selected on the basis of CSLS Score". Training the models in [6] using knowledge of POS tags along with words to obtain word embeddings has given comparable results for majority of cases.

Observing Figure 8 carefully would conclude that the word embeddings obtained from Cross-Lingual Word Embeddings(CLWE) model with POS tags are more uniformly distributed as compared to that of [6]. As for the visualisation of word embeddings we used the words with different POS tags, domains and fetched their nearest neighbour from the target language, hence they should not make clusters but be uniformly distributed with their translation in target language mapping close to the word in source language. This is visible from the PCA visualisation obtained from CLWE model with POS tags mentioned in Section 3.2.1 .

8 Conclusions and Future Work

In this work, we explored learning Cross-Lingual Word Embeddings using POS tags along with words. The idea of using dependency parse trees mentioned in Section 3.1 to resolve the issue of syntactic dissimilarity between languages has been experimented using graph based techniques i.e. node2vec and GraphSAGE. The experimentation using this technique is currently under progress. Hence the results are reported for the downstream task of word-alignment task using the model mentioned in Section 3.2.1 without using the technique mentioned in Section 3.1. The results obtained using this technique has given comparable results to [6] on word-alignment task for majority of cases. PCA word embeddings visualisation have shown that better quality embeddings is produced by CLWE with POS tags as compared to [6] by mapping the words with different domain more uniformly. In future we would like to complete the experiments using the technique mentioned in Section 3.1 .

References

[1] Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. Cross lingual word embeddings for low-resource language modeling. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

[2] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A survey of cross-lingual word embedding models. *arXiv:1706.04902*, 2019.

[3] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. *Proceedings of the 8th International Language Resources and Evaluation (LREC'12)*, 2012.

[4] Naveed Afzal, Ruslan Mitkov, and Atefeh Farzindar. Unsupervised relation extraction using dependency trees for automatic generation of multiple-choice questions. *Canadian AI 2011: Advances in Artificial Intelligence*, 2011.

[5] Mark Stevenson and Mark A. Greenwood. Dependency pattern models for information extraction. *Research on Language and Computation*, 2009.

[6] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Herve J egou. Word translation without parallel data. *arXiv:1710.04087v3*, 2018.

[7] Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing crosslingual distributed representations of words. *Proceedings of COLING*, 2012.

[8] Karl Moritz Hermann and Phil Blunsom. Multilingual distributed representations without word alignment. *arXiv:1312.6173*, 2013.

[9] Sarath Chandar A P, Stanislas Lauly, Hugo Larochelle, Mitesh M. Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. *arXiv:1402.1454*, 2014.

[10] Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. Transgram, fast cross-lingual word-embeddings. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[11] Stephan Gouws and Anders Søgaard. Simple task-specific bilingual word embeddings. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015.

[12] Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Learning crosslingual word embeddings without bilingual corpora. *arXiv:1606.09403v1*, 2016.

[13] Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. Massively multilingual word embeddings. *arXiv:1602.01925v2*, 2016.

[14] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv:1805.06297v2*, 2018.

[15] Takashi Wada, Tomoharu Iwata, and Yuji Matsumoto". Unsupervised multilingual word embedding with limited resources using neural language models. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[16] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. Question answering passage retrieval using dependency relations. *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005.

[17] Elif Aktolga, James Allan, and David A. Smith. Passage reranking for question answering using syntactic structures and answer types. *ECIR 2011: Advances in Information Retrieval*, 2011.

[18] Mihai Lintean and Vasile Rus. Paraphrase identification using weighted dependencies and word semantics. *Association for the Advancement of Artificial Intelligence (aaai)*, 2009.

[19] Vaishnavi V, Saritha M, and Milton R S. Paraphrase identification in short texts using grammar patterns. *2013 International Conference on Recent Trends in Information Technology (ICRTIT)*, 2013.

[20] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *arXiv:1607.00653*, 2016.

[21] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. *arXiv:1403.6652v2*, 2014.

[23] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014.

[24] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv:1607.04606v2*, 2017.