Cross Lingual Semantic Similarity

Mohit Choudhary, Shireen Nagdive, Bansari Patel
Stony Brook University, New York
{*mohit.choudhary,shireen.nagdive,bansari.patel*}*@stonybrook.edu*

Abstract

Translation of a word from one language to another is a very popular NLP Task. A lot of different approaches to this problem have one thing in common - the use of a large parallel corpora. In this project, we evaluate and improve a model that learns a simple mapping between vector spaces of two languages to fetch the top synonyms for a given word. We use an existing system that employs a small bilingual dictionary to achieve this task from English to Hindi. Given a word in English, we generate 10 most similar words for it in Hindi along with a similarity score for each of them. We further improve this model by utilizing a better bilingual dictionary and extending it to handle unseen words. We have also added support for - French, Bengali and Tamil. We build a bilingual dictionary for English- French, English-Bengali, English-Tamil and use this as a proof of concept that such a model would perform well given a good bilingual dictionary and a training corpus that encompasses a large number of words.

1 Introduction

1.1 Problem definition

The overall objective of this project is to evaluate a model that measures cross lingual semantic relatedness across Hindi and English words using word vectors. With the onset of globalization, information on the internet has emerged in a variety of languages. Due to a substantial growth of multilingual information and a growing demand for NLP applications for less common languages, it is necessary to have a model that easily maps relationships between words in different languages. However, it is difficult to obtain parallel corpora for training in languages that are not mainstream. A bilingual dictionary would be easier to create or come by. Establishing cross lingual relationships between languages directly benefits applications such as machine translation, cross-lingual information retrieval, cross-lingual text categorization and clustering, lexical substitution, question-answering, ontology alignment, plagiarism detection,etc. Hence, cross-lingual word semantic similarity measure is a very important and meaningful research topic.

1.2 Approaches to solve the problem

Distance based method : In this approach, the semantic similarity between two words is measured by considering the distance of the shortest path connecting the corresponding concepts in the taxonomy used. The basic idea of distance- based algorithms is to select the shortest path among all the possible paths between concepts of w1 and concepts of w2, given two input words w1, w2, and a lexical taxonomy. This approach assumes that the shorter the distance, the more similar the concepts are. [1], [2]

Information based method : Here the similarity is evaluated using external information, such as word frequencies and/or information content, extracted from available corpora, in addition to the hierarchical information related to the corresponding concepts in the underlying taxonomy. Information based methods were introduced to take advantage of the use of external corpora thus avoiding the unreliability of edge distances and taxonomies in the distance based approaches[3]

Recent approaches: The more recent approaches for capturing semantic similarity involves the use of distributed representations of words. Two particular models for learning word representations that can be efficiently trained on large amounts of text data are Skip-gram and

CBOW models. In practice, Skip-gram gives better word representations when the monolingual data is small. CBOW however is faster and more suitable for larger datasets[4].

1.3 Ideas for evaluation/extension of the baseline model

- From the initial runs, we noticed that the model performs well in most cases but the synonyms for certain words, especially human feelings are very inaccurate. We wanted to experiment with a more comprehensive training corpus to evaluate if it gives us a better result.

- We also want to evaluate the extent to which the size of the bilingual dictionary affects the model's performance. We believe that a small but evenly sampled bilingual dictionary should give us good results.

- The model generates top 10 synonyms in the target language for a given input word. Another evaluation metric is to take a pair of words in English as an input, generate their top synonyms and then compute the cosine distances between the pair of English words and the cosine distances between the pair of words in the target language. This gives us an idea of how closely the vector space for the two languages is mapped.

- The open source code we took as a baseline does not handle the case of unseen words - words. One possible extension to this would be to use NLTK to find the synonyms of this word and use the synonym to retrieve results from the model.

- We also want to extend this to encompass multiple target languages. So far, we have generated and tested models for French, Tamil and Bengali in addition to Hindi.

1.4 Evaluate

For evaluating our model, we measure the reduced MSE values of the original model and compare it with the MSE values after addition of new bilingual dictionaries.

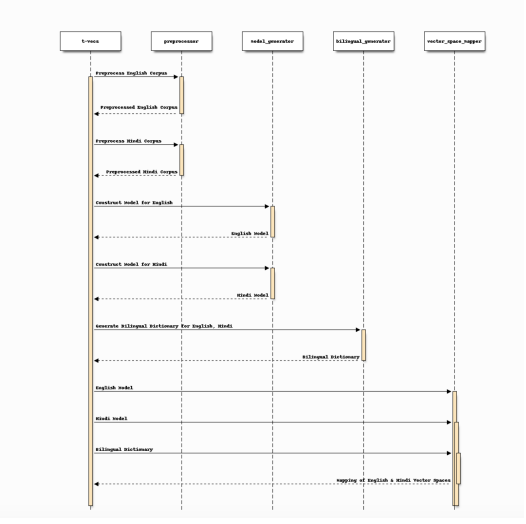
For synonym similarities, we use cosine score to evaluate the relatedness of word synonyms in two different languages.

2 Task

Our model for generating semantically similar words takes an input word in a given language and outputs 10 similar words in the target language. The first task is to choose a large enough text corpus in both source and target languages which encompasses a wide range of words. We will use these to build Word2Vec model. Now, even though we assume that the vector space for words across languages follow similar patterns, they will not have a perfect one to one mapping. There has to be some sort of an offset to these mappings. Our model has to learn this offset. In order to achieve this, we use a bilingual dictionary(explained later in the Section 3). We train our regression model to learn the mapping between vector spaces by feeding it vector lists for the set of W_s, W_t (word in source language, it's translation in target language). This model is then used to fetch the top 10 synonyms for any input word in the source language.

While our linear transformation approach makes use of a sampled bilingual dictionary, some standard solutions to this problem make use of parallel text as input and minimize the source and target language monolingual losses along with the cross-lingual regularization term.[5] Another approach to this problem involves the use of the word-level alignment of a seed bilingual dictionary to construct a pseudo-bilingual corpus by randomly replacing words in a source language corpus with their translations.[6]

2.1 Baseline Model(s)



The baseline model consists of 5 key components :

- Driver module for the entire model
- Modules for corpus pre-processing
- Model generator which creates vector spaces for the training corpora(using gensim's Word2Vec)
- Module for generation of bilingual dictionary
- Vector space mapper that learns the linear mapping between vector spaces of two languages and predicts the top synonyms.

2.2 Issues with Baseline Model

The baseline model generates word embeddings of size 100 and uses a fixed bilingual dictionary. Moreover, if a word that was not encountered in the training data is given as input, the model fails to predict the synonyms as it does not have a source language embedding to begin with. Moreover, the work has been done only for mapping from English to Hindi. extending this to multiple target languages would help establish a more concrete proof of concept for this particular approach. Our main ideas for extending and evaluating this baseline model have already been elaborated in Section 1.3 .

3 Approach

We start with the data corpus for the source and target languages which are basically large text extracts that encompass most of the dictionary. For the purpose of our project, we avoid using parallel corpora. Using gensims Word2Vec, we find the word-embeddings for the words in the training corpus using a skip-gram model. Once we have these two vector spaces, we need a model that will map the vector of an input word in the source language to similar words in the target language. We use a ridge regression model for this purpose. For the regression model to learn this mapping, we need to provide at least a small seed value of vector pairs for pairs of words in the source and target language. This is where the bilingual dictionary comes into the picture. If we used the entire training data to generate the bilingual dictionary, this would be no better than using a parallel corpus. Hence, we use a clustering approach to generate a sampled bilingual dictionary. Given the training data in the source language, we split it into similar clusters and from each cluster we pick a few words. For these words we find it's associated translation in the target language using Yandex[12]. We then loop over this dictionary which has W_s, W_t (word in source language, word in target language) and create vector lists of these words which are then fed to the regression model thereby helping it learn the mapping between vector spaces. The main intuition behind this approach is that similar words across languages will have similar vector space representations and so, learning a linear mapping between these two spaces should suffice. Hence, using a few words from each cluster of the source language as a seed would give us good results with limited amount of data. This approach forgoes the need for strictly parallel training corpora and the scripts to generate bilingual dictionary can be reused for multiple languages(we are using Yandexs translator found on GitHub to do the same).

3.1 Improving Word Embeddings

We experimented by increasing the embedding size to improve the vector spaces obtained. We also tried increasing the size of the bilingual dictionary to see if there was a significant improvement in the results. Another experiment was to use the pre-trained word vectors released by Facebook[11]. These models have been trained using enormous text corpora and give highly accurate results with a liner translation model. Owing to the fact that we did not have access to resources to train on such large data, we opted to use the pre-trained models to evaluate accuracy.

3.2 Handling Unseen Words

One of the drawbacks with the original model is it's inability to handle unseen words. We utilise NLTK for this purpose. For every word that was not seen in the training corpora, we utilize the Wordnet feature to find synonyms for these words. We then use our original model to find the translated words for the obtained synonyms. We do this till one of the synonyms return a result. Think of the word - Adumbrate. This is a rarely used word and will hardly be a part of the corpus which means that we would not a word-embedding value for it. However, the synonym of this word obtained through WordNet is very common - "outline", the translation for which can easily be found through our model.

3.3 Extension to Other Languages

The baseline model was limited to learning a linear mapping between English and Hindi. We extended this to include French, Tamil and Bengali as target languages. The intuition behind this is to both experiment and validate if such a simple model works well enough across multiple languages. From the three languages we experimented with, the accuracy obtained is reasonable enough.

3.4 Distance Comparison For Word Pairs

In order to evaluate if the vector space mappings across languages are reasonable enough for our model to learn linear mappings, one measure is to take a pair of words in English and find the cosine distance between them. We then find the top synonyms for this pair into a target language and find the cosine distance between the synonyms found. The offset between these distances is a measure of how varied the source and target language vector spaces are.

We wish to analyse just how effectively do synonyms translate from one language to another. Given two words in the source language which are synonyms, we find their corresponding closest translations. Using the cosine similarity score, we can find out the languages in which the translated synonyms are closest aligned to the source language.

4 Evaluation

4.1 Dataset Details

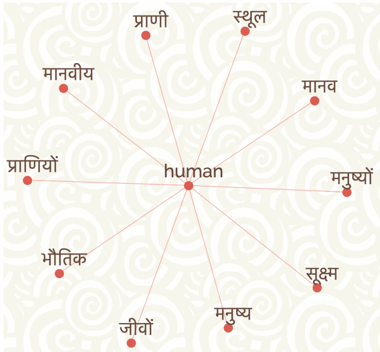
We have used Emille Corpora[7], Leipzig Corpora[8] and HCCorpora[9] to build word vector models for English and Hindi. Also, we have used Shabdakoshs[10] and Dicts Corporas[11] to build bilingual English - Hindi Dictionary.

4.2 Evaluation Measures

We measure the accuracy of our model by calculating the reduction in Mean Squared Error. The difference of the errors before and after the model training is the figure reported in the subsequent section. The higher the reduction, the better have the hyperparameters been tuned. We evaluate our results for both the word embeddings - one obtained through gensim Word2vec and the other released by Facebook. We report the reduction RMSE for these embeddings depending on which one has been used for the translation to a particular language. We then try different regression techniques to determine which one gives us the best result i.e the highest reduction in MSE. This comparison however has only been done for English to Hindi translation. For other languages, we simply use Ridge Regression which performs better than a vanilla linear regression model.

4.3 Results

Top 10 Hindi synonyms for the English word **human** (with similarity score):



मनुष्य	=>	0.75973880291
मानव	=>	0.721834897995
मानवीय	=>	0.709312856197
मनुष्यों	=>	0.692899227142
सूक्ष्म	=>	0.68816870451
प्राणियों	=>	0.684965014458
प्राणी	=>	0.679048836231
भौतिक	=>	0.676606535912
स्थूल	=>	0.667033553123
अदृश्य	=>	0.641482889652

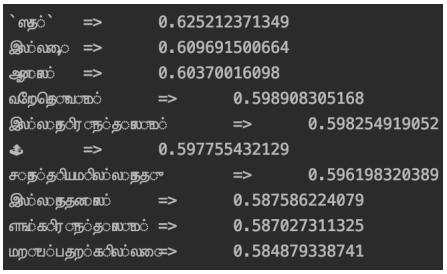
Top 10 French synonyms for the English word **love** (with similarity score):

tendresse	=>	0.678541779518
amour…	=>	0.66913163662
tendrement	=>	0.656502008438
aime	=>	0.649177193642
tristesse	=>	0.6459659338
amour	=>	0.64419811964
amoureux…	=>	0.643465280533
désir,	=>	0.635199069977
attendrissante	=>	0.632940530777
soupire	=>	0.627263188362

Top 10 Bengali synonyms for the English word **fish** (with similarity score):

ভোঁদড়	=>	0.784292459488
জলপোকা	=>	0.783092439175
ভোঁদড়ের	=>	0.782381057739
গাছফড়িং	=>	0.780858874321
শুঁয়োপোকা	=>	0.779929041862
পানকৌড়ি	=>	0.768510401249
পানমুরগি	=>	0.768008589745
মাছও	=>	0.766347825527
কাদাখোঁচা	=>	0.765982151031
ইঁদুরজাতীয়	=>	0.761520922184

Top 10 Tamil synonyms for English word **no** (with similarity score):



The distance between a pair of words in English is 0.37332 and the distance between their top synonyms in Hindi is 0.6264.

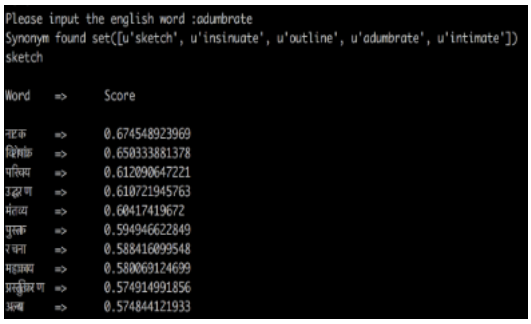
Similarly, the distance between a pair of words in English and the distance between their top synonyms in French :

Lang	Word1	Word2	Cosine distance
En	happy	cheerful	0.5497
Fr	heureux	enjouee	0.3758

Below are the Reduction MSE values for various values of embedding size, regression model, source and target language

Source Language	Target Language	Regression Model	EmbedingSize	Bilingual Dict Size	Reduction MSE	Execution Time(ms)
English	Hindi	Ridge	100	15057	55.5905	4001
English	HIndi	Linear	100	15057	30.2805	3989
English	Hindi	Ridge	300	38220	61.4289	27220
English	French	Ridge	300	113286	74.7164	41954
English	Bengali	Ridge	300	30737	67.9764	40933
English	Tamil	Ridge	300	26656	63.6927	41872

For unseen words, we return the top 10 similar words for its synonym.



4.4 Analysis

After increasing the word embedding size and using more accurate pre-trained word vectors, we saw an increase in reduction in root mean square error which means an improvement in performance. The model performs best with embedding size=300, 5 iterations and bilingual dictionaries that include one word per cluster for the training corpus.

In certain cases, the model retrieves words belonging to the same group as the input word rather than strictly synonyms. This is especially true in case of words that convey emotions and places/certain objects. Since we are returning the top 10 synonyms, this is bound to happen.

Ridge regression model gives a better accuracy. Mikolov[1] also achieved similar results in this aspect.

The distance between a pair of words in English and the distance between their top synonyms in the target language is a good measure of how closely the vector space between two languages correlates. We have used cosine distance for this evaluation and reported the result in the previous section.

4.5 Code

Original open source project used as baseline : https://github.com/KshitijKarthick/tvecs

Google drive link for our modified code : https://drive.google.com/file/d/1QPdl8FZ8EMUjLqcmoNmWWwdfpKzNtm9q/view

The project includes a README that has instructions on how to run the code. The model have already been trained and we have provided sample scripts for each of the functionality. The change in the codebase was made for multiple languages and maybe commented out in certain places. But for the purpose of evaluation and verification of the code and results, the scripts will be sufficient. The README mentions the changes.

5 Conclusions

A simple model that learns a linear mapping across vector spaces of the source and target language performs reasonably well given a sampled bilingual dictionary. From our experiments, we can conclude that on adding more words randomly to the bilingual dictionary, there is no drastic increase in accuracy. Hence, given random corpora in the source and target language, a limited bilingual dictionary helps us accomplish

the task. This eliminates the need of using parallel corpora where each word in the target language corpus has to be a strict translation of the word in source language corpus.

While the results obtained are not perfect(the most obvious translation might be the 4th best synonym), the top 10 synonyms mostly encapsulate related words. Using an embedding size of 300 for vectors drastically improves the accuracy. As an extension of the current work, we would want to experiment with different models and see how they compare against the Ridge Regression model.

6 Future Work

A possible extension to this project would be its use in search engines to parse articles across languages. For e.g if you are fluent in two languages and want your search query to return results in both of these languages, measuring the semantic similarity of keywords in articles in the secondary language could be a good measure to return these results.

References

[1] D. Yang and D. M. W. Powers. Measuring semantic similarity in the taxonomy of wordnet. In Proceedings of the 28th Australasian Computer Science Conference, pages 315 322, Newcastle,Australia, Jan/Feb 2005. Australian Computer Society.

[2] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In C. Fellbaum, editor, WordNet: An Electronic Lexical Database, pages 265283. MIT Press, 1998.

[3] P. Resnik. Using information content to evaluate semantic similarity in a taxoomy. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, volume 1, pages 448453, Montreal, Canada, August 1995.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[5] Tomas Kocisky, Karl Moritz Hermann, and Phil Blunsom. 2014. Learning Bilingual Word Representations by Marginalizing Alignments. In ACL.

[6] Xiao, M., Guo, Y. (2014). Distributed word representation learning for cross-lingual dependency parsing. In Proceedings of CoNLL, pp. 119129.

[7] http://www.emille.lancs.ac.uk/

[8] http://corpora.uni-leipzig.de/

[9] http://www.shabdkosh.com/content/category/downloads/

[10] http://dicts.info/dictlist1.php?l=Hindi

[11] https://github.com/facebookresearch/fastText/blob/master/pretrained vectors.md

[12] https://translate.yandex.com/