Creating Sense Vectors using Cross-lingual data

Rachita Chhaparia[1], Deepanshu Gupta[2]

Advisor: Dr Amitabha Mukherjee[1*]

[1] *Department of Computer Science and Engineering, IIT Kanpur*

[2] *Department of Mathematics and Statistics, IIT Kanpur*

E-mail: {rachitac,dpanshu,amit}@iitk.ac.in

Abstract

In this project we propose a novel approach to the problem of Word-Sense Disambiguation i.e. the task of automatic identification of the sense of a polysemous word in a given context. This project exploits bilingual features to overcome problem of accidental polysemy i.e. identification of coarse sense of word. We provide a novel approach that creates and uses sense vectors to approach the task of WSD using unsupervised learning. Our results show the promise in this approach given its simplicity and future extensions.

1. Introduction

The most important feature of Word Sense Disambiguation is to predict the sense of word, in which it is used, given the context space of the word. It has been looked over by generations of researchers, both linguists and non-linguists. Occasional tasks like Senseval and Semeval gather together researchers on WSD platform. Note the famous quote by J. R. Firth "You

[*] To whom correspondence should be addressed

shall know a word by the company it keeps". Firth's concept of a word's relevance is highly correlated with the task of Word Sense Disambiguation setting.
Notice how the change the context of word **plant** change its semantic meaning in the following example.

- No one here knew how to **plant**$_1$ crops, but the next town over was made up of farm laborers.

- The mining and washing **plant**$_2$ is extremely good and largely constructed at Cagliari.

- Tulsi **plant**$_3$ is well respected and worshipped in India.

Note that in the first sentence the word **plant** refers to placing something to grow, whereas in second example we have a reference to an industrial site and the third example refers to flora.

A related problem, described below, is a single representation of these polysemous words in the word vector space.

In the vector space setting, words of similar semantics are distributed closely. However, within such a setting, we often find that the word space distribution is polluted owing to the different meanings and uses of certain words, called polysemous words, in different contexts. One solution is to create a separate vector for each sense of a polysemous word. These vectors are called sense vectors.

In this project, we formulate a novel cross-lingual approach to create sense vectors for use in WSD.

To disambiguate in a language (source), using another less related language (target;having different proto-languages) can prove to be a useful resource as a polysemous word in the source language will likely translate to distinct un-related words in the target language. Hence the different translations, in turn the different vector representations, can be employed to differentiate between the coarse senses of word.

This report in Section 2 explains some earlier approaches to WSD, Section 3 explains our approach, and provides with our resources & tools and their pre-processing. Section 4 goes on to our experimental results. Section 5 suggests future improvements and extension to our approach and we conclude this report with Section 6.

2. Previous work in WSD

Extensive amount of research has been done in WSD. A few of the important related works are:

2.1 Using Raw Counts in comparable corpora[1]

Ranking sense distributions using the raw counts of translations of a word in target language to another language.They used Expectation Maximization based formulation to determine the sense frequencies. They then tagged each word with the most probable sense without giving any particular weight or importance to context of appearance of the word.

2.2 Multi-Prototype Vector-Space Models[2]

They initialize the number of meanings of a word without taking into consideration the word. Then the different vectors are randomly distributed in the vector space. Using clustering techniques the word senses are clustered to produce groups of similar context vectors. Then they use these clusters to determine the semantic similarity of different words.

2.3 Sense-specific Word Embeddings using parallel corpus[3]

Guo et al. represent words with multiple and sense-specific embeddings, which are learned from bilingual parallel data(English-Chinese).They then perform word similarity measurement to capturing the sense-level word similarities. Their intuition in using this approach is same as ours that ”Same word in the source language with different senses is supposed to have different translations in the foreign language”.

2.4 Unsupervised word sense tagging using parallel corpora[4]

They identify words in target language and their corresponding translation in source language. Then target language words are grouped that translate to same word in foreign language. Each sense is considered in this group and using semantic similarity sense tags are selected for the source word. Our approach is similar to this one with the change that we utilize rich vector representations instead of translation to tag senses.

3. Our Approach, Resources and Toolboxes

3.1 Our Approach

Assumptions

In our project, we disambiguate in English and use Hindi as a secondary language. Our method uses two assumptions:

- Word vectors of similar words in two different languages are related by a linear transformation[5]

  This transformation matrix (W) that relates word vectors in two languages (source,target) can be learned by using a bilingual dictionary D containing top 5000 most frequent words in the source language as the training set and performing stochastic gradient descent to minimize the following function:

  $\min_W \sum_{i=0}^{n} \|Wx_i - z_i\|$, where $(x_i, z_i) \in D$, $x_i \in \text{vectorSpace}_{source}$, $z_i \in \text{vectorSpace}_{target}$

  We use Hindi as the source language and English as our target language, such that given the vector representation, v_{Hindi} of a word in Hindi, Wv_{Hindi} gives a vector in the English word vector space, such that they are semantically similar.

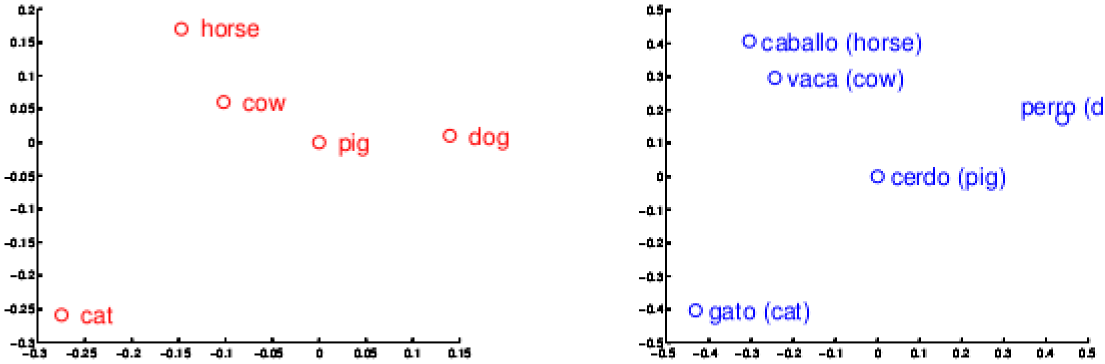


Figure 1: Projected distributed word vector representations of numbers and animals in English (left) and Spanish (right), taken from[5]

- A polysemous word in one language (here, English) translates to a set of distinct un-related words in another language (here, Hindi)

  For example, in the above example, $plant_1$ is ugana, $plant_2$ is karkhaana, $plant_3$ is podha in Hindi, three distinct and un-related words.

Methodology

1. **Word Vectorisation**:

   English: Pre-trained vectors (about 1B words) using Mikolov's Word2Vec with CBOW architecture. The model contains 300-dimensional vectors for 0.2 million words.

   Hindi: Vectors trained on HindMonoCorp0.5 using Mikolov's Word2Vec with skip-gram architecture. The model contains 300-dimensional vectors for 0.07 million words.

2. **Learning the Translation Matrix**: A linear mapping i.e. a translation matrix $W_{h \to e}$ from the Hindi word-vector space to the English word-vector space was learned using stochastic gradient descent using a bilingual dictionary consisting of most frequent 5000 words in the Hindi corpus for training.

3. **Sense Translation**: The senses of 30 polysemous words in English were translated into Hindi using online Google Translate. For example,$\{$ $plant_1 - ugAnA, plant_2 - kArkhAnA, plant_3 - podhA\}$.

4. **Context Vectorization**: A context window of size, say 'k' (a parameter) was taken around the target word and their vectors were averaged to form the context vector, We also, down-weighted the vectors based on their frequency in the corpus and their distance from the target word in the context.

   v_i be the context vector for i^{th} word and w_i be the word vector of i then :

   $v_i = \sum_{j=-k,j\neq i}^{k} w_j$ where k are context window words

5. **Sense Vectorization**: For every English polysemous word, the vectors (x) of its translated senses in Hindi are mapped back to English $(W_{hi\rightarrow en}x)$ which are the corresponding sense vectors for the word in English. For example, $W_{hi\rightarrow en}v_{ugana}$, $W_{hi\rightarrow en}v_{karkhaana}$, $W_{hi\rightarrow en}v_{podha}$ are respective sense vectors for plant in English, where v_x represents the word vector of word x in Hindi.

6. **Sense Disambiguation**: For every target word, we calculate the cosine similarities between the context vector and each of its sense vectors created above. The sense whose vector has the maximum cosine similarity, denoted by *csim* is chosen as the correct sense, where:

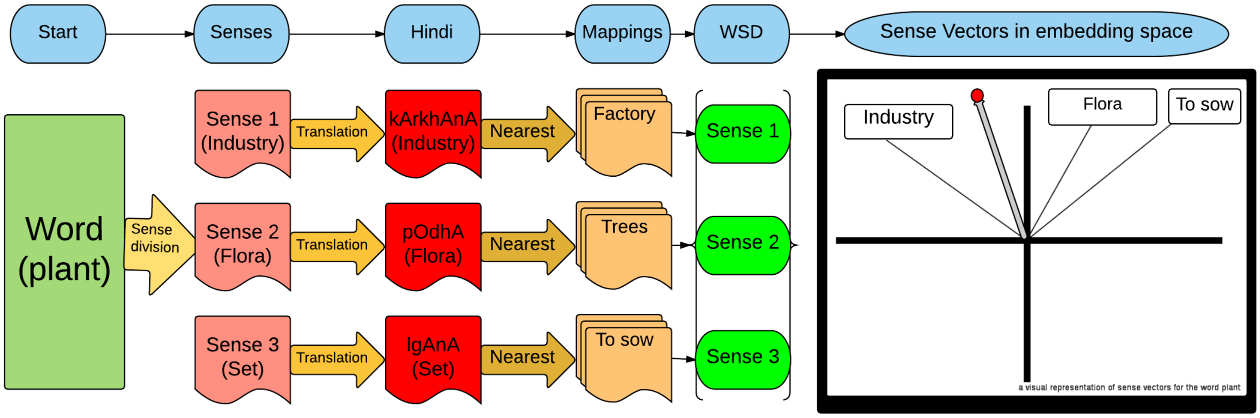
   $csim(\overrightarrow{(}u), \overrightarrow{(}v)) = \frac{\overrightarrow{(}u).\overrightarrow{(}v)}{\|u\|\|v\|}$

   For example in the example we provided we expect that

   $csim(\overrightarrow{(}plant_2), \overrightarrow{(}context)) > csim(\overrightarrow{(}plant_1), \overrightarrow{(}context))$

   $csim(\overrightarrow{(}plant_2), \overrightarrow{(}context)) > csim(\overrightarrow{(}plant_3), \overrightarrow{(}context))$

   The following figure illustrates each step for the above example of plant:



3.2 Resources and Toolboxes

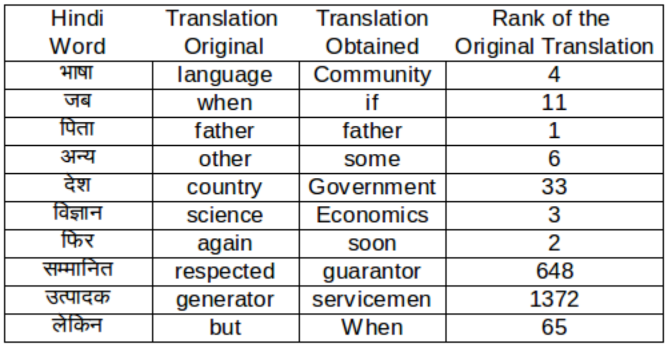
We utilize Word2Vec[6] to generate 300 dimensional word embeddings for Hindi and use Google News vectors[5] available open source. We also utilize 64 dimensional word embeddings available from polyglot project[7] for preliminary testing purposes. For testing data set purposes we utilize data available from Senseval-1.[8] To generate corresponding Hindi Senses we use Hindi word net from CFILT, IIT Bombay.[1]

4. Results

A lot of the results of our work depend on the validity of the translation matrix. We utilized the Hi-En dictionary provided by CFILT, IIT Bombay to test our translation accuracy. We tested across 200 most frequent words available in it. We obtained the following results for predicting the Hindi-English translation pair in the evaluation:

Precision @ 1	**Precision @5**	**Precision @ 10**
5.5%	13.5%	22.5%

Also note some of the translations that we obtained from our training:



Hindi Word	Translation Original	Translation Obtained	Rank of the Original Translation
भाषा	language	Community	4
जब	when	if	11
पिता	father	father	1
अन्य	other	some	6
देश	country	Government	33
विज्ञान	science	Economics	3
फिर	again	soon	2
सम्मानित	respected	guarantor	648
उत्पादक	generator	servicemen	1372
लेकिन	but	When	65

Note that in the translations that we obtain we get collections of word that are used in quite similar contexts. Thus our assumption of linear transformation of vector spaces.

However a point to note is that even though a source word's sense translate to distinct words in target language. There might be a problem that the sense might have many translation all of them belonging to the same synset. Thus calculating accuracy of translation using just one word is not a good approach.

Hence the translation part is very crucial in our final task also. Because a lot will depend on word that the sense is mapped against. Should the word change we might get a new structure of output. For example for the first sense of **accident** when ”tkkr” was used we got higher accuracy as compared to ”bhidnA” even though both mean the same.

The test set for the final evaluation on WSD task was taken from Senseval-I, consisting of approximately 200 instances each of 30 polysemous words. The synsets were made coarse-grained manually. The average accuracy across all instances was **42%** with a standard deviation of **18%**. Some examples:

Phrase	**Correct**	**Prediction**
unable to work because of unemployment, illness or an **accident**.	crashmod	crashmod
they have the **sanction** of the New Zealand Rugby	econ-action	econ-action
we extricated the **scraps** of crisp batter	morsel	waste
Hitting an industry when it is nearly on its **knees**	bow	patella

- The accuracy for a word depends on the sense translations made. The more distinct they are, the better is the accuracy. For example, for the word 'knee', consider two senses 'bow' and 'patella'. They are used in almost the same context in both languages and hence, cannot be disambiguated accurately (around 50%)
- Changing the way we calculate the context vectors has a positive impact on the prediction task. We can also explore other context measure to improve the WSD predictions.

Accuracies for some polysemous words:

Words(#coarse senses)	**Accuracy**(simple average)	**Accuracy** (sweighted average)
giant (5)	90%	91%
amaze (3)	88%	95%
promise (3)	72%	73%
derive (3)	55%	56%
sanction (3)	55%	55%
slight (3)	58%	59%
modest (2)	36%	37%
bet (5)	15.3%	15.6%
sack (6)	2.3%	2.7%
accident(3)	67.3%	67.9%

5. Future work

We have in our project tried for "single word" - word vectors. We believe one can extend this project to phrasal embeddings and have very improved amount of accuracy.

Another extension of the project would be to look at integration of our technique with POS tagging approach(which is quite popular in WSD tasks). During our testing and translation times we noticed the problem of Hubness in the vector space i.e. that some words appear close to each word in the vector space. If one can get over this problem of Hubness, it shall shoot high the accuracy. Also, improving the translation matrix can provide a boost to the method. Either way this approach has very high potential in sense that there are a lot of improvements that can boost the performance of WSD task and help in eliminating the problem of Accidental Polysemy. One can also try different corpora - Comparable corpora or Parallel corpora to train the initial embedding. This shall guarantee that the embeddings are very similar in nature and the most frequent words align more or less perfectly.

6. Conclusion

We have presented in this project a new approach to WSD using sense vectors created from bilingual resources. This approach has shown quite commendable results (42% average accuracy) given that it is very simple and there's a lot of room for improvement. Also, this approach can be applied to any pair of languages hence, can be used to disambiguate in any language.

7. Acknowledgement

We thank Prof. Amitabha Mukherjee for his constant guidance and feedback that helped in improving this project. We also thank our friends whose regular suggestions and motivation were a boost to us.

References

(1) Bhattacharyya, S. B. S. S. P. Neighbors Help: Bilingual Unsupervised WSD Using Context. Proceedings of the Conference. 2013; p 538.

(2) Reisinger, J.; Mooney, R. J. Multi-prototype vector-space models of word meaning. Hu-

man Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. 2010; pp 109–117.

(3) Guo, J.; Che, W.; Wang, H.; Liu, T. Learning sense-specific word embeddings by exploiting bilingual resources. Proceedings of COLING. 2014; pp 497–507.

(4) Diab, M.; Resnik, P. An unsupervised method for word sense tagging using parallel corpora. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. 2002; pp 255–262.

(5) Mikolov, T.; Le, Q. V.; Sutskever, I. *arXiv preprint arXiv:1309.4168* **2013**,

(6) Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. *arXiv preprint arXiv:1301.3781* **2013**,

(7) Al-Rfou, R.; Perozzi, B.; Skiena, S. *arXiv preprint arXiv:1307.1662* **2013**,

(8) Kilgarri, A. Senseval: An exercise in evaluating word sense disambiguation programs. Proc. of the first international conference on language resources and evaluation. 1998; pp 581–588.