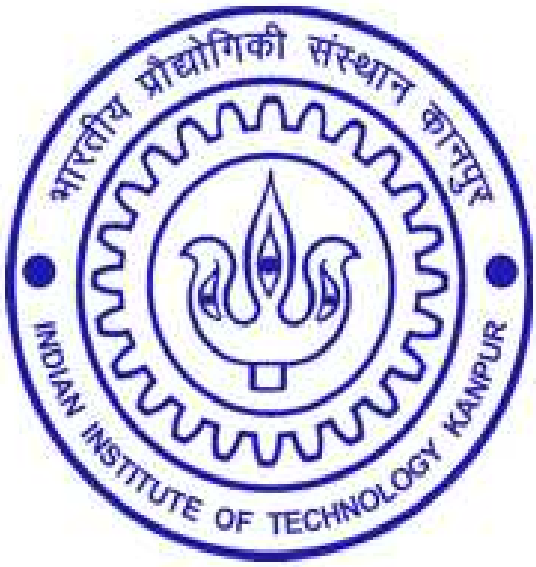
Discovering Lexical Classes and Syntax using Word Embeddings (Report with Annotated Bibliography)

Pranjal Singh(10327511)

November 25, 2014



Purpose: M.Tech. Thesis(CS699)
Semester I(2014-15)
Advisor: Prof. Amitabha Mukerjee

Abstract

Word embeddings have the power to capture semantics.They have potential to represent syntax and semantics both. We have many sources of unsupervised raw data but not supervised data. Unsupervised techniques could greatly improve existing supervised (Collobert et al.(2013)). By leveraging large amount of data floating around, we can improve existing systems. We want to design a framework that will be unsupervised and can induce lexical classes and syntax from the given data by building vector representations, also popularly known as word embeddings. We may further want to extend this framework for languages such as Hindi which is poor in terms of resources.

1 Problem Statement and Idea

LSA and LDA were used to capture word embeddings(not exactly) and hence derive semantic relations. Most of the existing systems treat word as atomic units but words also inherit meanings which can only be defined if we represent it as a vector/combination of latent words. So the goal is to maximize probability of raw text given a context window. So for a given context window of size c:

$$\max \frac{1}{T} \sum_{t=1}^{T} \log p(w_t | w_{t-c}^{t+c})$$

2 Work Done

- Survey of existing literature
- Ran code provided by Collobert and Weston[9] (SENNA)
- Buit vector space representation of Hindi words by training on wikipedia dump
- Derived similar words by using Cosine Similarity technique to create visual embeddings

3 Presentation

- Semester presentation
  http://home.iitk.ac.in/ spranjal/thesis/bmg_sem9.pdf

4 Earlier Work

This section discusses three important earlier work related to the current work.

4.1 word2vec

This work utilized two architectures discussed below:

4.1.1 CBOW

Embeddings are represented by a set of latent variables and initialized randomly. Training learns these for each word w_t in the vocabulary.
So for a given context window of size c:

$$\max \frac{1}{T} \sum_{t=1}^{T} \log p(w_t | w_{t-c}^{t+c})$$

where

$$\max p(w_t | w_{t-c}^{t+c}) = \frac{exp({e'}_{w_t}^{T} \cdot \sum_{-c \leq j \leq c, j \neq 0} e_{w_{t+j}})}{\sum_w exp({e'}_{w_t}^{T} \cdot \sum_{-c \leq j \leq c, j \neq 0} e_{w_{t+j}})}$$

4.1.2 Relational Constraint Model

Define R as a set of relation between two words and relations have scores associated to indicate strength.

$$\max \frac{1}{N} \sum_{i=1}^{N} \sum_{w \in R_{w_i}} \log p(w|w_i)$$

4.2 NLP from Scratch

This work proposes a unified architecture for NLP tasks such as POS tagging, Chunking, NER and SRL. They have compared against classical NLP benchmarks. The works avoids task specific engineering and generalizes the system to handle multiple tasks.
They learn lookup table by back propagation. The words are mapped to d-dimensional vector using lookup table operation. At the end, the lookup table returns a matrix for a given sentence which can be used in training the neural network.
They have used entire English Wikipedia to learn word embeddings (631 million words) and they tokenized it using Penn Treebank tokenizer.
The total training time was about four weeks. The parameters of the model were: Window size: 11 and a Hidden layer with 100 units
The objective is to seek a network that computes a higher score when given a legal phrase than when given an incorrect phrase.

$$\theta := \sum_{x \in X} \sum_{w \in D} max\{0, 1 - f_\theta(x) + f_\theta(x^{(w)})\}$$

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
454	1973	6909	11724	29869	87025
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

Figure 1: Word Embeddings

4.3 Bilingual Word Embeddings

It proposes a method to learn bilingual embeddings rather than just monolingual embeddings. So it utilizes counts of Machine Translated alignments derived from Berkeley aligner to initialize monolingual embeddings of another language.

$$W_{t-init} = \sum_{s=1}^{S} \frac{C_{ts}+1}{C_t+S} W_s$$

They have used the same formulation as *Collobert et al.*(2008) to learn embeddings except that they have used global context information as in *Huang et al.*(2012)

Their objective function captures information of both monolingual embedding and

also on translation matrices, also called alignment matrices. They have trained on 100K-vocabulary word embeddings. With 500,000 iterations it took 19 days of training on 8-core machine. For phrase similarity in 2 languages, they have averaged out the word embedding vectors corresponding to each word in both phrases and then taken cosine similarity to quantize amount of semantic similarity.

5 Dataset

- English wikipedia dump (Size: 95MB)
- Hindi wikipedia dump (Size: 279MB)

6 Results

6.1 English

6.1.1 Embeddings

We find the top 10 embeddings for the word *girl* given below.

"boy" is to "father" as "girl" is to ...?

Word	Cosine Similarity
Mother	0.6219688653945923
Grandmother	0.5560075640678406
Wife	0.5442352890968323

6.1.2 Similarity

This experiment finds the most similar word for the query.

Given Word11	Given Word12	GivenWord21	FoundWord22
he	his	she	**her**
big	bigger	bad	**worse**
going	went	being	**were**

6.1.3 Odd One Out

This experiment finds the odd word in the provided list.

breakfast cereal dinner lunch

Result: *cereal*

6.2 Hindi

6.2.1 Embeddings

We find the top 10 embeddings for the word *bharat* given below.

6.2.2 Odd One Out

This experiment finds the odd word in the provided list.

Figure 2: Top few Embeddings in context of another word

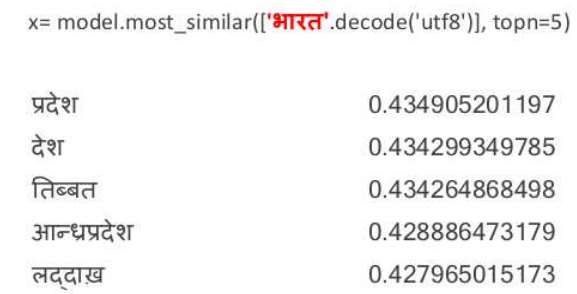


Figure 3: Top 5 embeddings for word *bharat*

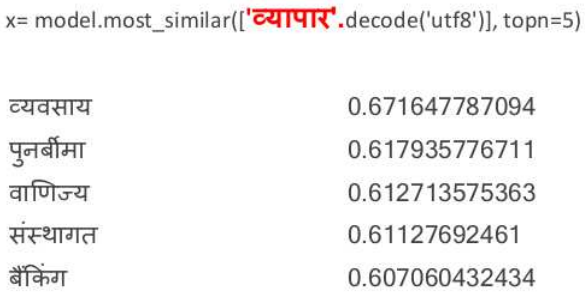


Figure 4: Top 5 embeddings for word *vyapar*

Figure 5: Blue is the odd one out; Red ones are the queries

7 Future Work

Collobert-Weston claim that their system is generalized for various NLP tasks. This could lead to a big disaster if we use these embeddings for Sentiment Classification

task. The current trainig conditions would bring words such as **good** and **bad** in close proximity. But they tend to represent opposite poles. This is an open area to extend the work.
One exciting think to look at is what if we use these embeddings as functions and operate on other embeddings, e.g., if we ADD the embeddings or if we SUBTRACT the embeddings.

Very Big
Bigger

Such phrases and words should have greater semantic similarity. We could experiment with operations such as addition/subtraction and could get a better insight into such relationships (applicable for Hindi also).

Indian Cricketer
Sachin

Infact above phrase and word may belong to same embedding because they they tend to be very close when it comes to their semantic relationship.
The embeddings obtained could help in initializing the embeddings used in work of Collobert and Weston where they suffered problems due to random initialization.
Manning et al.(2013) have used semantic information to improve word embeddings and *Collobert et al.*(2008) have used large unlabeled data to do the same thing. We could use syntactic or morphological information to improve word embeddings or even produce some good word embeddings. The motivation for this idea could be that morphologically similar words have some sought of close connection between them.
e.g. morphology, phonology, etymology

8 Acknowledgement

I would like to show my sincere gratitude towards Prof. Amitabha Mukerjee, Computer Science and Engineering Department, IIT Kanpur for his motivation to do this work and becoming my advisor for this work. I would also like to thank my all friends who have helped me throughout this project.

References

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

> The recently introduced continuous Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships. In this paper the authors present several extensions that improve both the quality of the vectors and the training speed. By subsampling of the frequent words we obtain significant speedup and also learn more regular word representations. We also describe a simple alternative to the hierarchical softmax called negative sampling. An inherent limitation of word representations is their indifference to word order and their inability to represent idiomatic phrases. For example, the meanings of Canada and Air cannot be easily combined to obtain Air Canada. Motivated by this example, they present a simple method for finding phrases in text, and show that learning good vector representations for millions of phrases is possible. A very interesting result of this work is that the word vectors can be somewhat meaningfully combined using just simple vector addition. Another approach for learning representations of phrases presented in this paper is to simply represent the phrases with a single token. Combination of these two approaches gives a powerful yet simple way how to represent longer pieces of text, while having minimal computational complexity.

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

> They propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. They observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, they show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.The main goal of this paper is to introduce techniques that can be used for learning high-quality word vectors from huge data sets with billions of words, and with millions of words in the vocabulary. The first proposed architecture is similar to the feedforward NNLM, where the non-linear hidden layer is removed and the projection layer is shared for all words (not just the projection matrix); thus, all words get projected into the same position (their vectors are averaged). They call this architecture a bag-of-words model as the order of words in the history does not influence the projection.
>
> The second architecture is similar to CBOW, but instead of predicting the current word based on the context, it tries to maximize classification of a word based on another word in the same sentence.

[3] Will Y. Zou, Richard Socher, Daniel M. Cer, and Christopher D. Manning. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398. ACL, 2013.

They introduce bilingual word embeddings: semantic embeddings associated across two languages in the context of neural language models. They propose a method to learn bilingual embeddings from a large unlabeled corpus, while utilizing MT word alignments to constrain translational equivalence. The new embeddings significantly out-perform baselines in word semantic similarity. A single semantic similarity feature induced with bilingual embeddings adds near half a BLEU point to the results of NIST08 Chinese-English machine translation task. It utilizes counts of Machine Translated alignments derived from Berkeley aligner to initialize monolingual embeddings of another language. They have used the same formulation as *Collobert et al.*(2008) to learn embeddings except that they have used global context information as in *Huang et al.*(2012)
Their objective function captures information of both monolingual embedding and also on translation matrices, also called alignment matrices. They have trained on 100K-vocabulary word embeddings. With 500,000 iterations it took 19 days of training on 8-core machine. For phrase similarity in 2 languages, they have averaged out the word embedding vectors corresponding to each word in both phrases and then taken cosine similarity to quantize amount of semantic similarity.

[4] Omri Abend, Roi Reichart, and Ari Rappoport. Improved unsupervised pos induction through prototype discovery. In Jan Hajic, Sandra Carberry, and Stephen Clark, editors, *ACL*, pages 1298–1307. The Association for Computer Linguistics, 2010.

In this work, the authors present a novel fully unsupervised algorithm for POS induction from plain text. This is inspired by cognitive notion of prototypes. Prototype theory is a mode of graded categorization in cognitive science, where some members of a category are more central than others.
This algorithm first identifies landmark cluster of words, which are essentially the prototypes, and then maps rest of the words to these clusters. The work is evaluated on English and German where authors utilize morphological and distributional representations computed in fully unsupervised manner.
Central members define the category. The algorithm is as follows:
1) First cluster words based on fine morphological representation.
2) Cluster most frequent words, define landmark clusters.
3) Maps the rest of the words to these clusters.
For distributional representation, left and right context is taken into account and then scores are defined on this basis.
For clustering, they use average-link clustering. The comparison is done with a tagged corpus. The measures used to evaluate the model are mapping-based measures(many-to-one and one-to-one) and information theoretic measure(these are based on the observation that a good clustering reduces the uncertainty of the gold tag given the induced cluster, and vice-versa).
The authors say that punctuation marks are very frequent in corpora and are easy to cluster. As a result, including them in the evaluation greatly inflates the scores. For English the model was trained on 39832 sentences from section 2-21 of PTB-WSJ AND ON 500K sentences from NYT section of NANC newswire corpus. There are 45 clusters in this annotation scheme, 34 of which are not punctuation.
They used k=14 and k=34 for running the algorithm.

[5] Michael Lamar, Yariv Maron, Mark Johnson, and Elie Bienenstock. Svd and clustering for unsupervised pos tagging. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 215–219, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Abstract
They revisit the algorithm of Schutze(1995) for unsupervised part-of-speech tagging. The algorithm uses reduced-rank singular value decomposition followed by clustering to extract latent features from context distributions. As implemented here, it achieves state-of-the-art tagging accuracy at considerably less cost than more recent methods. It can also produce a range of finer-grained taggings, with potential applications to various tasks.

Method
The new SVD-based approach also termed as "two-step SVD" or"SVD2" has four important characteristics. First, it achieves state-of-the-art tagging accuracy. Second, it requires drastically less computational effort than the best currently available models. Third, it demonstrates that state-of-the-art accuracy can be realized without disambiguation, i.e., without attempting to assign different tags to different tokens of the same type. Finally, with no significant increase in computational cost, SVD2 can create much finer-grained labellings than typically produced by other algorithms.
The authors construct a left and a right context matrix to take into account the successor and predecessor of a given token. They take top 1000 most frequent words in the matrices. SVD2 is used to create descriptors and then k-means clustering is used to create groups.

Data and Results
Full Wall Street Journal part of the Penn Treebank(1,173,766 tokens) was used and capitalization was ignored. Evaluation was done against the POS-tag annotations of the 45-tag PTB tagset and against Smith and Eisner coarse version of the PTB tagset. Three evaluation criteria of Gao and Johnson(2008): M-to-1, 1-to-1 and VI were used.
The performance of SVD2 compares favorably to the HMM models. With $k_2 = 45$, SVD scores 0.659 in accuracy on PTB45.

[6] Joseph Turian, Dpartement Dinformatique Et, Recherche Oprationnelle (diro, Universit De Montral, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *In ACL*, pages 384–394, 2010.

In this work, authors have tried to improve existing systems for NER and Chunking by inserting unsupervised word representations as extra word features into them. They have evaluated Brown clusters, Collobert and Weston (2008) embeddings, and HLBL embeddings(Mnih & Hinton, 2009). They have also tried with various combinations of word representations.
The goal of the work is to learn word features. They have classified word representations into Distributional, Cluster-based and Distributed.
Distributional word representations are based upon a co-occurrence matrix F of size $W \times C$, where W is the vocabulary size, each row F_w is the initial representation of word w, and each column F_c is some context. Here, context types can be left window, right window, or size of window and frequency count can be raw, binary, or tf-idf. LSA and LDA are based on this model of word representation.
Clustering-based word representations induce a clustering over words. Brown algorithm, which is a hierarchical clustering algorithm, clusters words to maximize information of bigrams. In hierarchical clustering, we can choose the word class at several levels in the hierarchy, which can compensate for poor clusters of a small number of words.
A distributed representation is dense, low-dimensional, and real-valued. Distributed word representations are called word embeddings. Each dimension of the embedding corresponds to a latent feature of the word. These are typically induced using neural language models. Collobert and Weston (2008) and HLBL embeddings are based on this model.
The authors have used unsupervised data for inducing word representations. They have classified word representations as a mathematical object associated with each word, often a vector.
They have followed conditions in the CoNLL-2000 shared task for chunking (Sang & Buchholz, 2000) for chunking. They have used CRFsuite by Naoaki Okazaki. The data for chunking is from Penn Treebank, and is newswire from Wall Street Journal in 1989. For NER, they have used regularized average perceptron model. The standard evaluation benchmark for NER is the CoNLL03 shared task dataset drawn from the Reuters newswire.
The authors compare to the state-of-the-art methods of Ando and Zhang (2005), Suzuki and Isozaki (2008), and for NERLin and Wu (2009).
Brown clusters perform best for both NER and chunking. Also mixture of two models give quite good result as expected. C&W embeddings outperform the HLBL embeddings.

[7] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *In Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL*, 2012.

Abstract

Unsupervised word representations are very useful in NLP tasks both as inputs to learning algorithms and as extra word features in NLP systems. However, most of these models are built with only local context and one representation per word. This is problematic because words are often polysemous and global context can also provide useful information for learning word meanings. The authors present a new neural network architecture which 1) learns word embeddings that better capture the semantics of words by incorporating both local and global document context, and 2) accounts for homonymy and polysemy by learning multiple embeddings per word. The authors introduce a new dataset with human judgments on pairs of words in sentential context, and evaluate our model on it, showing that their model outperforms competitive baselines and other neural language models.

Data and Results

the authors evaluate their model on the standard WordSim-353 dataset that includes human similarity judgments on pairs of words, showing that combining both local and global context outperforms using only local or global context alone, and is competitive with state-of-the-art methods.

By utilizing global context, this model outperforms C&Ws vectors and certain baselines on this dataset. With multiple representations per word, the authors show that the multi-prototype approach can improve over the single-prototype version without using context. Moreover, using AvgSimC4 which takes contexts into account, the multi-prototype model obtains the best performance.

[8] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *CoRR*, abs/1003.1141, 2010.

Abstract

Computers understand very little of the meaning of human language. This profoundly limits our ability to give instructions to computers, the ability of computers to explain their actions to us, and the ability of computers to analyse and process text. Vector space models (VSMs) of semantics are beginning to address these limits. This paper surveys the use of VSMs for semantic processing of text. The authors organize the literature on VSMs according to the structure of the matrix in a VSM. There are currently three broad classes of VSMs, based on term-document, word-context, and pair-pattern matrices, yielding three classes of applications. The authors survey a broad range of applications in these three categories and we take a detailed look at a specific open source project in each category. The goal in this survey is to show the breadth of applications of VSMs for semantics, to provide a new perspective on VSMs for those who are already familiar with the area, and to provide pointers into the literature for those who are less familiar with the field.

[9] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398, 2011.

Abstract

This paper proposes a unified neural network architecture and learning algorithm that can be applied to various natural language processing tasks including: part-of-speech tagging, chunking, named entity recognition, and semantic role labeling, achieving or exceeding state-of-the-art performance in each on four benchmark tasks. Our goal was to design a flexible architecture that can learn representations useful for the tasks, thus avoiding excessive task-specific feature engineering (and therefore disregarding a lot of prior knowledge). Instead of exploiting man-made input features carefully optimized for each task, the system learns internal representations on the basis of vast amounts of mostly unlabeled training data. This work is then used as a basis for building a freely available tagging system with excellent performance while requiring minimal computational resources.