ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/106478, December 2016* ISSN (Online) : 0974-5645

Word Embedding Models for Finding Semantic Relationship between Words in Tamil Language

S. G. Ajay*, M. Srikanth, M. Anand Kumar and K. P. Soman

Centre for Computational Engineering and Networking (CEN), Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amrita University, Coimbatore - 641112, Tamil Nadu, India; ajay190694@gmail.com, srikanthmurali3@gmail.com, m_anandkumar@cb.amrita.edu, kp_soman@amrita.edu

Abstract

Objective: Word embedding models were most predominantly used in many of the NLP tasks such as document classification, author identification, story understanding etc. In this paper we make a comparison of two Word embedding models for semantic similarity in Tamil language. Each of those two models has its own way of predicting relationship between words in a corpus. **Method/Analysis:** The term Word embedding in Natural Language Processing is a representation of words in terms of vectors. Word embedding is used as an unsupervised approach instead of traditional way of feature extraction. Word embedding models uses neural networks to generate numerical representation for the given words. In order to find the best model that captures semantic relationship between words, using a morphologically rich language like Tamil would be great. Tamil language is one of the oldest Dravidian languages and it is known for its morphological richness. In Tamil language it is possible to construct 10,000 words from a single root word. **Findings:** Here we make comparison of Content based Word embedding and Context based Word embedding models respectively. We tried different feature vector sizes for the same word to comment on the accuracy of the models for semantic similarity. **Novelty/Improvement:** Analysing Word embedding models for morphologically rich language like Tamil helps us to classify the words better based on its semantics.

Keywords: CBOW, Content based Word Embedding, Context based Word Embedding, Morphology, Semantic and Syntactic, Skip Gram

1. Introduction

In this paper we deal with the concept of word similarity and its associative models. Similarity in words can be understood by finding feature vectors for each word. Using these vectors ensures understanding of words more than their syntactic or semantic regularities. Word similarity is taken from the models namely Content and Contextual based Word embedding[1]. The results give us the semantic and syntactic information of the words respectively. Semantic similarity is one of the trivial concepts in natural language processing, cognitive science, artificial intelligence, psychology. Word similarity measurements in terms of semantics are used in many practical applications like information retrieval, synonym extraction, clustering of documents. The syntactic similarity is nothing but order of words in a sentence with comparison to the vector for each word. For example, *vector (king - man + woman)* is close to *vector (queen)*[2]. We can even compute past and present tense of words and draw relation from it. This approach is known as syntactic word similarity. The Continuous Bag of Words and Skip gram model captures semantic as well as syntactic regularity. So we use them in Content and Contextual based Word embedding for finding word similarity. Word similarities help us understand the working principle of these models. Our aim is to study the semantic relationship between words using Word embedding. Semantic similarity tells about how words are semantically related. For example, the word pair *cars* and *gasoline* are more related than the word pair *cars* and *bicycles* but the latter pair is actually related. The Content based Word embedding model gives the words which are semantically similar. So far Latent Relational Analysis[3] proposed by[3] is the best

* *Author for correspondence*

method for finding semantic similarity. This approach is an extension of conventional Vector Space Models[4] for finding semantic similarity[5]. Thus semantic similarity has got applications in automatic meaning identification, Ontology creation, keyword identification and word analogy. Natural Language Processing tools like POS-tagger, Morphological generator[6], Machine Translation System[7] for Tamil language have been developed, so by incorporating the proposed models into these tools, results with better accuracy can be obtained.

2. Mathematical Background

Word embedding models used here is not a single monolithic algorithm. It consists of two unique models such as CBOW (Continuous Bag of Words) and Skip gram. We use CBOW and Skip gram models of Content and Contextual based Word embedding respectively. Figure1 and Figure 2 explain the Architecture of CBOW and Skip gram model respectively.

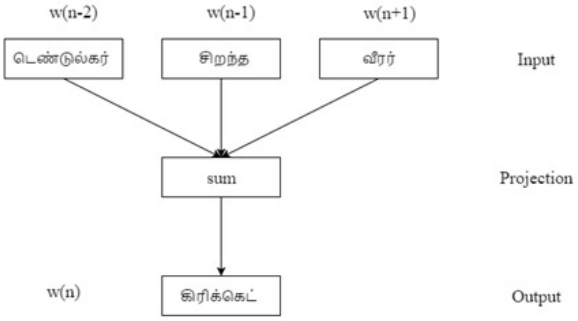

Figure 1. Architecture of CBOW model.

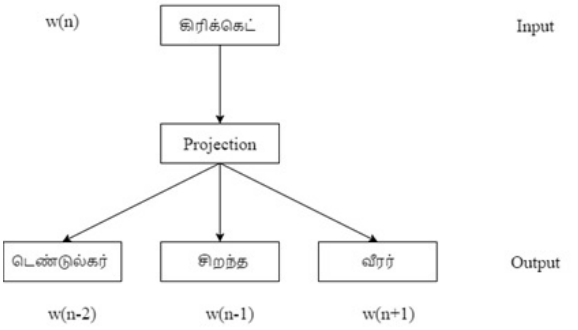

Figure 2. Architecture of skip gram model.

The model complexity trade for efficiency can give us better word representations. They are two layered neural networks because it is trained to reconstruct linguistic context of words[8]. They take input as a large context of data and produce vectors in space, which contains several dimensions. A unique word from corpus will have a unique vector in space. The vectors of the word are positioned at vector space in such a manner that words which are sharing common contexts in the dataset are in closer proximity compared to other words[2]. Word embedding used here is quite an algorithm which preserves the word order information. Syntactic tasks can be managed by having word order preservations. Content based Word embedding model lacks in word order preservation which is ensured by the Contextual Word embedding model. The Content based Word embedding model are insensitive to word order resulting in partial optimal results for POS-Tagging and dependency parsing, while the other model is sensitive to word order. The two models are understood based on these CBOW and Skip gram models. The Skip gram's functionality is to increase the likelihood of the prediction of contextual words given the centre word[9]. Consider T as the number of words in the given document. We maximize the likelihood L,

$$L = \frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-c\le j<c \\ j\neq 0}} logp(w_{t+j}|w_t) \quad (1)$$

Where c is the hyper parameter defining the window of context words.

$$p(w_0|w_i) = \frac{e^{0^{w_i}}(w_0)}{\sum_{w\in v} e^{0^{w_i}}(w)} \quad (2)$$

Where $p(w_0|w_i)$ is the output probability. The word w_0 is the predicted word, given the word w_i. This is nothing but softmax approach, for more vocabularies we use hierarchical softmax objective. The CBOW model is able to predict the centre word w_0 given the surrounding words[9] w_{-1} and w_{+1}.

3. Methodologies of Word Embedding based Similarity

3.1 Content based Word Embedding model using Neural Networks

We implement CBOW and Skip gram models of Content based Word embedding using Gensim. Gensim has an

inbuilt language model named word2vec[9] (written in python) which works based on neural networks[10]. So the model is imported and trained using the data set. The model is trained with input parameters such as size (which mentions the vector dimensions). After the model is trained with the input parameters a new model is generated. The model is saved permanently by giving it a name and can be used anytime. The generated model is tested with input words from the corpus. For a given input word we can find out its top 10 most similar words from the corpus. These most similar words give an idea about the content of the document. For CBOW two models are created with different feature vectors sizes of 100, 200 i.e., each word is mapped to 100 and 200 dimensional space respectively, similarly this process is repeated with same parameters for Skip gram. The results produced by these four models for a single word is tabulated and compared. The reason why we go for different feature vector sizes is that increasing feature vector dimension may improve accuracy in the results.

3.2 Context based Word Embedding Model

We implement CBOW and Skip gram models of context based word embedding using *gcc* compiler in Linux operating system preferably Ubuntu. The library files of this model are open source and it is written in C language. This model is implemented using *wang2vec*[11], an extension of *word2vec* using different architectures[10]. In addition to CBOW and Skip gram it also has some architecture like *cwindow* and *Structured Skip gram*. But in our paper we make a comparison of CBOW and Skip gram models only. The implementation procedure is similar to the former model. Here instead of creating a model file as in the former model we create a binary file (.bin) as output. It is similar to model file and can be used to find word analogy. For a given input word we can find out its top 10 most similar words from the corpus. These most similar words give an idea about the context in which the word is present. For CBOW two models are created with different feature vectors sizes of 100, 200 i.e., each word is mapped to 100 and 200 dimensional space, similarly this process is repeated with same parameters except for Skip gram. The results produced by these four models for a single word is tabulated and compared. Thus 8 models are created (4 from CBOW and 4 from Skip gram).

3.3 Clustering of Words based on its Similarity

In order to understand the working of the above mentioned models, we interpret the result in a different way. The words or feature vectors obtained using the Content based Word embedding model are clustered. Clustering plays an important role in dimensionality reduction. By treating each word cluster as a single feature, dimensionality can be drastically reduced. This preserves classification accuracy when compared to the normal classifier. Clustering is essential for NLP tasks like Paraphrase detection, Named Entity Recognition[12,13].

4. Results and Discussion

The data set we use is a collection of political news articles of India from various newspapers in Tamil language. Our data set is huge and it has around 2.7 lakh sentences which contains 50 lakh words. So by using this huge data set or corpus which covers words of all classes (open and closed) like noun, verb, adjective, preposition, determiners etc., we will be able to draw a conclusion which gives an insight of the best of both models. Four words (*Jayalalitha, Sethu, District and Chennai*) in Tamil language are taken from the corpus and word similarity is found using the above mentioned 8 models.

word2vec				wang2vec			
CBOW		Skip gram		CBOW		Skip gram	
Feature Vector Dimension size							
100	200	100	200	100	200	100	200
ஜெயலலிதாவின்	ஜெயலலிதாவின்	முதல்வர்	முதல்வர்	ஜெயலலிதாவின்	ஜெயலலிதாவின்	முதல்வர்	முதல்வர்
ஜெயலலிதா	ரங்கசாமி	முதலமைச்சர்	ஜெயலலிதா	ஜெயலலிதாவால்	ஜெயலலிதாவால்	முதல்வர்	ஜெயலலிதா
சித்தராமையா	என்.ரங்கசாமி	ஜெயலலிதா	ஜெயலலிதா	ஜெயலலிதா	ஜெயலலிதா	முதலமைச்சர்	ஜெயலலிதா
ரங்கசாமி	ஜெயலலிதா	முதல்வர்	முதல்வர்	ஜெயலலிதா	ஜெயலலிதா	ஜெயலலிதா	முதல்வர்
என்.ரங்கசாமி	ஜெயலலிதாவால்	ஜெயலலிதா	முதலமைச்சர்	ஜெயலலிதாவிடம்	ஜெயலலிதாவிடம்	அறிவித்தார்	முதலமைச்சர்

Figure 3. Similar words from the corpus for the word (Jayalalitha).

word2vec				wang2vec			
CBOW		Skip gram		CBOW		Skip gram	
Feature Vector Dimension Size							
100	200	100	200	100	200	100	200
சமுத்திரத்	சமுத்திரத்	சமுத்திரத்	சமுத்திரத்	சமுத்திரத்	சமுத்திரத்	சமுத்திரத்	சமுத்திரத்
சமுத்திர	சமுத்திர	சமுத்திர	சமுத்திர	சமுத்திர	சமுத்திர	சமுத்திர	சமுத்திர
இணைப்புத்	இணைப்புத்	கால்வாய்த்	கால்வாய்த்	கால்வாய்த்	கால்வாய்த்	கால்வாய்த்	கால்வாய்த்
கால்வாய்த்	கால்வாய்த்	சமுத்திரக்	சமுத்திரக்	சமுத்திரக்	சமுத்திரக்	சமுத்திரக்	சமுத்திரக்
சேதுசமுத்திரத்	சேதுசமுத்திரத்	சேதுசமுத்திரத்	சேதுசமுத்திரத்	சேதுசமுத்திரத்	சேதுசமுத்திரத்	சேதுசமுத்திர	சேதுசமுத்திர

Figure 4. Similar words from the corpus for the word (Sethu).

word2vec				wang2vec			
CBOW		Skip gram		CBOW		Skip gram	
Feature Vector Dimension Size							
100	200	100	200	100	200	100	200
மாவட்டம்	மாவட்டம்	வட்டம்	வட்டம்	மாவட்டம்	மாவட்டம்	மாவட்டம்	மாவட்டம்
மாவட்டத்தில்	மாவட்டம்	மாவட்டம்	மாவட்டம்	வட்டம்	வட்டம்	வேப்பந்தட்டை	வட்டம்
மாவட்டம்	வட்டம்	வனச்சரகம்	வனச்சரகம்	மாவட்டம்	மாவட்டம்	வட்டம்	வேப்பந்தட்டை
மாவட்டத்திலுள்ள	பென்னாகரம்	கிராமத்தைச்	ஆண்டிப்பட்டி	செங்கிப்பட்டி	மாவட்டம்	தேவகோட்டை	மல்லசமுத்திரம்
வட்டம்	மாவட்டத்திலுள்ள	விருதாச்சலம்	தம்மம்பட்டி	விருதாச்சலம்	விருதாச்சலம்	வலங்கைமான்	வலங்கைமான்

Figure 5. Similar words from the corpus for the word (District).

word2vec				wang2vec			
CBOW		Skip Gram		CBOW		Skip Gram	
Feature Vector Dimension Size							
100	200	100	200	100	200	100	200
மதுரை	மதுரை	சென்னை	சென்னை	எழும்பூர்	சென்னை	சென்னை	சென்னை
சென்னையில்	சென்னையில்	கோவை	சைதாப்பேட்டை	சைதாப்பேட்டை	சைதாப்பேட்டை	மதுரை	நுங்கம்பாக்கம்
சென்னை	கோவை	சைதாப்பேட்டை	நுங்கம்பாக்கம்	சென்னை	எழும்பூர்	கோவை	மதுரை
கோவை	திருச்சி	எழும்பூர்	எழும்பூர்	கோடம்பாக்கம்	மீனம்பாக்கம்	சைதாப்பேட்டை	சென்னையில்
திருச்சி	எழும்பூர்	மதுரை	மீனம்பாக்கம்	நுங்கம்பாக்கம்	கோடம்பாக்கம்	மீனம்பாக்கம்	சைதாப்பேட்டை

Figure 6. Similar words from the corpus for the word (Chennai).

ID NO	Words in the cluster
ID 498	சி.டி.செல்வம், ராஜேஷ்குமார், பால் வசந்த குமார், சதாசிவம், டி.குன்ஹா
ID 445	ஆடிபெருக்கை, அம்மனை, கிரிவலம், சபரிமலை, திருவிழா
ID 315	கூறியவர், கூறியவர்கள், கூறியுள்ளன, கூறியும், கூறுவார்கள்
ID 102	அ.இ.அ.தி.மு.க, அ.இ.அ.தி.மு.கவிற்கு, காங்கிரஸ், பா.ஜ.க, ம.தி.மு.க
ID 256	திருமாவளவன், தொல் திருமாவளவன், ஞானதேசிகன், மு.க.ஸ்டாலின், கருணாநிதி

Figure 7. Clustering of words from the corpus based on its similarity.

The results are shown above. From Figure 3 to Figure 6 we see that Content based Word embedding model produces better results (similar words) based on the semantic regularity, whereas Contextual based Word embedding model produces better results based on the syntactic regularity[11]. Next we created 500 clusters using feature vectors of each word obtained by Content based Word embedding model. We give each cluster a unique cluster ID. The Figure 7 gives an insight of the results obtained using clustering. We took randomly 5 clusters from the 500 clusters and mentioned in the result. Each cluster ID'S contain semantically similar words based on the content. Since Content based Word embedding model is unsupervised approach based on Neural Networks it is able to draw semantically similar words from the corpus and form clusters. **ID 498** is a cluster which contains the names of all judges in India. **ID 445** contains all the words which name the religious rituals, temple names etc. **ID 315** contains words which denote all verb forms of a speech. **ID 102** contains the names of political parties in India. ***ID 256*** contains names of all political leaders in India.

5. Conclusion and Future Work

In this paper, two models namely Content and Contextual Word embedding are described in terms of their performances obtained from the semantic similarity experimental result. The semantics is obtained from the Content based Word embedding, whereas syntactics is obtained from Context based word embedding. This work can be extended to find word analogies as analogy

between words are hidden in the word embedding. So by finding word analogies the accuracy of document classification, information retrieval can very well be increased. This work can also be extended to find Cross Lingual Semantic Similarity. Using this we can find words which are similar in meaning from different languages. This plays an important role in multilingual Machine Translation System.

6. References

1. Goldberg Y, Levy O. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722; 2014.
2. Bengio Y, Schwenk H, Senecal J-S, Morin F, Gauvain J-L. Neural Probabilistic Language Models. JMLR. 2006; 137–86.
3. Turney P. Measuring semantic similarity by latent relational analysis. Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence; 2005.
4. Turney PD, Pantel P, et al. From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research. 2010; 37(1):141–88.
5. Resnik P. Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007; 1995.
6. Rekha RU, Kumar MA, Dhanalakshmi V, Soman KP, Rajendran S. A novel approach to morphological generator for Tamil. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2012; 6411: 249–51.
7. Kumar MA, Dhanalakshmi V, Soman KP, Rajendran S. Factored statistical machine translation system for English to Tamil language. Pertanika Journal of Social Sciences and Humanities (JSSH). 2014; 22(4):1045–61.
8. Mikolov T. Statistical language models based on neural networks. Presentation at Google; Mountain View. 2012 Apr 2.
9. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. arXiv:1310.4546v1; 2013. p. 3111–9.
10. Mnih A, Teh YW. A fast and simple algorithm for training neural probabilistic language models. arXiv preprint arXiv:1206.6426; 2012.
11. Ling W, Dyer C, Black A, Trancoso I. Two/too simple adaptations of word2vec for syntax problems. Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies; 2015.
12. Abinaya N, John N, Ganesh HBB, Kumar MA, Soman KP. AMRITA-CEN@FIRE-2014: Named entity recognition for Indian languages using rich features. ACM International Conference Proceeding Series; 2014 Dec 05-07. 103–11.
13. Abinaya N, Kumar MA, Soman KP. Randomized kernel approach for named entity recognition in Tamil. Indian Journal of Science and Technology. 2015; 8(24).