Semi-Supervised Word Sense Disambiguation for Telugu Exploiting Word Embeddings and Monosemous Synonyms

Srinivas Mulkalapalli[1], Padmaja Rani B[2]

[1]Research Scholar, JNTUH, Hyderabad, Telangana, India
[2]Professor,CSE Department, JNTUH, Hyderabad, Telangana, India

ABSTRACT

In every natural language there are some ambiguous words having multiple meanings but only one of them is appropriate in the sentence. Determining the appropriate sense of an ambiguous word in sentence that fits to the context is known as Word Sense Disambiguation(WSD). WSD is not a standalone problem, but it part of several other Natural Language Processing(NLP) tasks. In this paper, we propose semi-supervised WSD for Telugu language based on word embeddings and sense embedding obtained through substitution of monosemous synonyms of each sense of an ambiguous word. Popular word embeddings tool word2vec is used to compute the embeddings of words in the large text corpus created from the Telugu Wikipedia dump, Telugu News Articles, and Telugu books. Continuous-Bag-of-Words (CBOW) model is used since it is faster than Skip-Gram model. It results in only one vector for each word without considering the multiple senses for ambiguous words. Sense specific embeddings which are necessary for WSD are obtained by first substituting monosemous synonyms for each sense of every ambiguous word. Then, corresponding sense vector is obtained by taking the mean of word vectors of synonyms. Context vector is obtained from the content words surrounding the target ambiguous word. Similarity score is computed between context vector and each sense vector of the target ambiguous word. The sense vector resulting in highest similarity score is returned as the correct sense of an ambiguous word. We have evaluated our WSD on a test dataset that consists of 1500 Telugu sentences for 100 most frequently used ambiguous words in Telugu and achieved an accuracy of 76.53%.

Keywords:Natural Language Processing,Word Sense Disambiguation, Word Embeddings, word2vec, Telugu

INTRODUCTION

In natural language processing, resolving ambiguity is one important research problem. Ambiguous words, having multiple senses or meaning, is prevalent in almost every language used by human race to communicate their opinion or ideas. Telugu language, spoken in South India, also has many ambiguous words. Human beings are very skillful in resolving ambiguity if arises during their communication using their world knowledge. But, for computers the task of finding the most appropriate sense of an ambiguous word in a sentence is very difficult. It is even considered as one of AI Complete problem. The task of determining the most appropriate sense of an ambiguous word implied through the context in a sentence or domain is known as Word sense Disambiguation (WSD). WSD is one of the important research areas where development of good systems helps to improve the performance of several NLP applications like Machine Translation, Information Retrieval, and similarity analysis etc. Substantial research was completed and in continuation for English and other European languages because of the availability of required lexical resources such as WordNet [1], Corpus [2] and Golden Dataset for Evaluation [3] and very less work has been done for Telugu language [4, 5].

The research on WSD for Indian languages such as Hindi, Tamil, Bengali, Telugu, and Malayalam… etc. is hampered due to unavailability of well-established lexical resources such as WordNet, Corpus etc. But, Indo-WordNet [6] is a lexical database developed for the Indian languages by Pushpak Bhattacharya et al. and providing access through http://www.cfilt.iitb.ac.in/indowordnet/index.jsp. Telugu language also has many ambiguous words. Recently, with the development of ICT many NLP Tasks such as Sentiment Analysis, Text Classification, Question- Answering Systems, Information Retrieval, Machine Translation etc. for Telugu are also in need. WSD for Telugu is not matured enough to be useful in the above mentioned tasks. But, WSD for other languages like English, German, French, Italy, Chinese is well developed because of sophisticated lexical resources and corpus. So, we are interested in developing good WSD systems for Telugu in hope of being useful in several NLP tasks.

RELATED WORK

A. WSD Techniques

Lesk's algorithm [7] is one in determining the most appropriate sense of an ambiguous word implied from its context. It proceeds as follows: A separate bag of words is created for every possible sense of the target word from its definition found in Machine Readable Dictionaries. They have used Oxford Advanced Learner's Dictionary of English. Similarly, a context bag is created from all other words surrounding the target word taking their definitions. The algorithm determines the number of common words between each sense bag and the context bag. Finally, the sense whose sense bag has highest common words is selected as the appropriate meaning. Later, glosses form WordNet are used for Word Sense Disambiguation in [8]. They have achieved an accuracy of 32% by evaluating their algorithm through participation in SENSEVAL2, which involves an evaluation exercise on English lexical sample data. Knowledge based WSD for Hindi is proposed in [9]. Prity Bala has evaluated the proposed system on a dataset of 100 ambiguous words and got 50% accuracy. Alok Ranjan Pal et al in [10] proposed knowledge based WSD using Bengali WordNet. They shared the results obtained by conducting experiments on dataset containing 9 frequently used Bengali ambiguous words.

Dependency parsing [11] and word embeddings [12] are also plays crucial role for resolving disambiguate for words. A knowledge-based approach to WSD exploiting Structural Semantic Interconnection is presented in [13] by Roberto Navigli. WSD based on conceptual density is proposed in [14] by Agirre and Rigau. Conceptual density is computed from is-a hierarchy from the WordNet. C. Leacock and M. Chodorow proposed a WSD based on combining local context and WordNet similarity in [15]. D. Lin explored the possibility of using syntactic dependency to improve WSD in [16]. Semantic distance between topics in WordNet is used for Word Sense Disambiguation by Sussan in [17]. Sussan proposed a weighting scheme based on WordNet relations. The synonymy relation is assigned a weight of zero, whereas the weights in the range [1, 2] are assigned to other relations hypernymy, hyponymy, holonymy and meronymy. WSD for Hindi based on measure of Semantic Relatedness is proposed by Satyendr Singh et al in [18]. They have achieved 60.65% as an overall average accuracy by conducting an experiment on a sense tagged dataset prepared by them which consists of several instances for 20 polysemous Hindi nouns. Knowledge based WSD for Telugu is proposed in [19] by SuneethaEluri and VishalaSiddu. They have achieved an accuracy of 65.4 by evaluating on Telugu sentences formed for 150 ambiguous words both nouns and verbs.

B. Word Embeddings

In single prototype vector, we have a unique representation for word irrespective of its several possible senses. For WSD, we require a different vector for each sense. Detailed survey on various approaches is presented in depth in [20]. So, lot of research was conducted to avoid this limitation. We briefly present some of them here. Sense induction is achieved through explicit pre-clustering of contexts using TF_IDF in [21]. Centroid of resulting clusters represents the sense prototype. But, resulting sense prototypes do not correspond to word meaning.

Neural networks are used to learn vector representations of words by Huang et al. in [22]. Word embeddings for context words are obtained and later they are clustered. It is followed by relabeling of word occurrences in the corpus to which they belong to. Retraining is done to get the sense embeddings. In count based language models, we use the context to get the matrix of word co-occurrences WC with the rows and columns correspond to words in the context which are ordered some way. A value of k in the entry of row i and column j denotes that word i and word j occur k times together in the context. These matrices tend to be sparse, so require dimensionality reduction to save space and retain the quality. In predictive neural network models, we learn word vectors through training a neural network which works on predecessors to predict the next word. Earlier models were impractical because of more computational complexity. Later, Mikolov et al. made it practical by proposing simpler architecture in [23] that require linear computation only.

C. Telugu Resources

To develop WSD for Telugu, we require two Telugu resources: A large raw text corpus to get the word embeddings and a lexical knowledge database(LKB) of monosemous synonyms for each sense of ambiguous word.

Text corpus

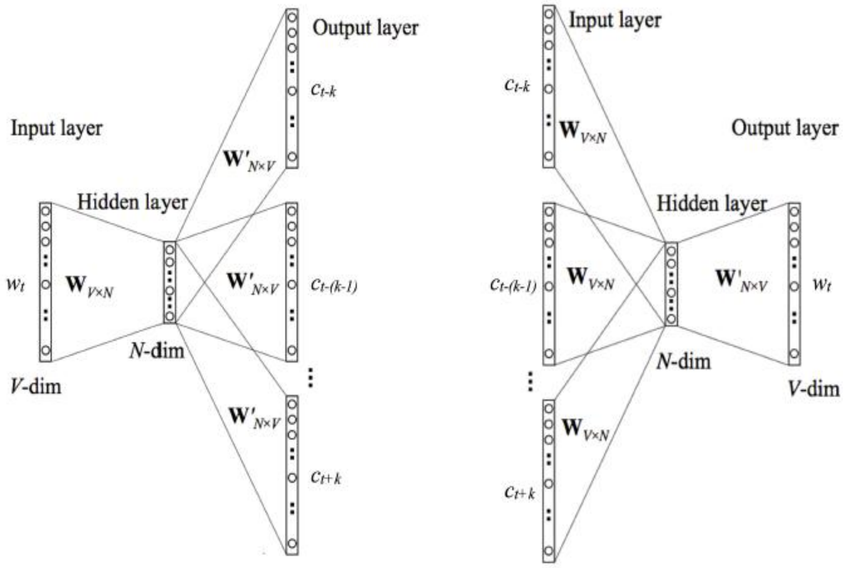
We witness the availability of large amount of data for all languages on the web. It allows us to collect very large text corpus with little effort which is fundamental and necessary resource for all corpus-based studies required while developing various NLP models. Producing quality embeddings require large-scale text corpus which is discussed in [24]. We have extracted Telugu Wikipedia dump available for public. Additionally, it is augmented by downloading Telugu News articles, Telugu books. We have preprocessed the corpus that involves eliminating unnecessary information, other language text, replacing digits by the symbol NUM to get clean and meaningful text corpus to be used in several NLP tasks. Stop words are removed from the above corpus. We got the Telugu stop words from TDIL, India.

Telugu Lexical Resources

Inspired by the role of English WordNet in developing various NLP tasks in English, WordNet for other languages are being developed. Telugu WordNet is available along with other 17 languages in IndoWordNet. But, as Telugu WordNet is in the early stages of development, we could not find complete information required to provide monosemous synonyms of for each sense of an ambiguous word. We have collected additional information from Telugu Nighantuvu for 30 ambiguous words selected for evaluation by us.

Word Vectors Initialization

We have used word2vec toolkit [25] to learn word vectors of the words in the text corpus created in 2.3.1. word2vec is neural generative predictive model using deep learning techniques. It produces a dense vector representation for words which successfully captures syntactic and semantic features of words. Word2vec follows the principle of distributional hypothesis which so that words occur and used in similar context are assigned similar vector representations. Word2vec model consists of 2 architectures: Continuous-Bag-of-Words (CBOW) and Skip-Gram. CBOW predicts the target word based on given context words, whereas Skip-Gram predicts the context words based on target word. Order of words has no impact on the operation of CBOW. Here, we preferred CBOW as it is faster compared to Skip-Gram. The architectures of word2vec models is as shown in Fig. 1.



a.CBOW Model b. Skip-Gram model

Figure 1: Architectures of word2vec models

We trained CBOW word embeddings with 300 dimensions, size of the context window as 4 and setting minimum word frequency as 5. We evaluate the accuracy of word vectors by using them in the task of word semantic similarity. Telugu word semantic similarity dataset is prepared by us that consists of 50 Telugu pair of words whose similarity is in the scale of 1 to 10. We got an accuracy of 94%.

Sense Embeddings

The disadvantage of word embeddings generated through word2vec is that it produces a unique vector for each word including ambiguous words which have multiple meanings. This makes the word embeddings not suitable for WSD. We need some way of obtaining sense embeddings for ambiguous words. Replacing the ambiguous word by the monosemous synonyms of the implied sense does not change the correctness of the sentence. Based on this, we devised a procedure to compute the sense vectors of each ambiguous word from the monosemous synonyms.

Suppose that w_i is the vector of word type i, and $w_{ij^{|}}$ j=1,…n are vectors of monosemous synonyms of w of particular sense i, i=1,…N. Then, the corresponding sense vector is obtained as follows:

$$\mathrm{W_{ij}} = \frac{\mathrm{W_i} + \sum_{\mathrm{j^{|}=1}}^{\mathrm{n}} \mathrm{W}_{\mathrm{i}j^{|}}}{\mathrm{n}+1}$$

The word vectors inventory is augmented with sense embeddings as a separate inventory.

Semi-Supervised Telugu Word Sense Disambiguation(SSTWSD):

Our method uses the word embeddings computed through word2vec and sense-specific embeddings obtained in the previous section. Context vector is computed based on the content words surrounding the target word. Similarity is computed between the context vector and each sense vector of the target word using cosine similarity. It returns the sense with the highest similarity as the answer. We describe the procedure in detail below:

Computing Context Vector

It is computed taking the average of all word embedding vectors forming the context. Suppose that C is the word list, |C| is the size off the wordlist, vec(.) is a function returning the word embedding of argument computed by word2vec. Then, the following equation will give the context vector

$$\mathrm{CV} = \frac{1}{|C|}\sum_{i=0}^{|C|} vec(c[i])$$

Cosine Similarity

The cosine similarity computed between any two vectors lies in the range [0,1]. The values close to 1 indicate that the corresponding vectors are semantically similar while the values close to 0 manifest that the corresponding vectors are semantically unrelated. It is computed using the following equation.

$$sim_{w2v}(w_i, w_j) = \frac{w_i . w_j}{\|w_i\| \, \|w_j\|}$$

where w_i and w_j are two words or terms.

Algorithm: Pseudocode of proposed method SSTWSD

Input: A sentence S, a target word w, word embeddings WE, Sense Inventory SI

Output: Correct sense of the target word w

Procedure:

```
Step 1: Tokenize the sentence S. Remove stop words and perform lemmatization to get the list of content words W.
Step 2: Form a context bag C from W ignoring the target word w.
Step 3: Get the context vector CV from C using the vectors corresponding to the words in C from WE
Step 4: Retrieve the sense vectors V = {$v_1$, $v_2$, …, $v_n$} of target word w from the sense inventory SI. n is the number of senses of target word w.
Step 5: Result ← empty; score ←0;
Step 6: for each $v_i$ in V:
            Cscore ← CoSim(CV, $v_i$)
            if (score <cscore):
score ← cscore
result ← $v_i$
Step 7: Return $v_i$, Sense
```

RESULTS

It is evaluated against the test dataset that consists of 1500 sentences for 100 most frequently used ambiguous words. The Evaluation measure of Accuracy is used to evaluate our algorithms.

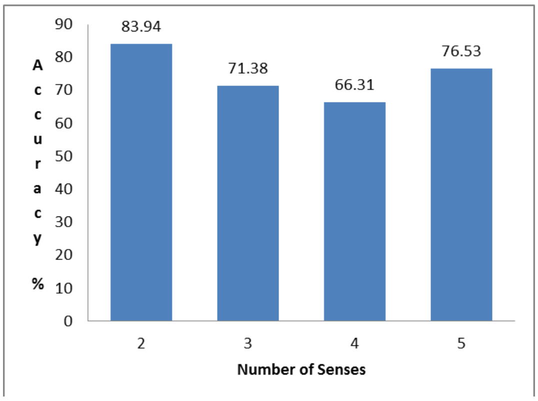


Figure 2: Results of SSTWSD Algorithm for words

$$\text{Accuracy} = \frac{\#\,\text{Test Instances correctly disambiguated}}{\text{Total Instances of ambiguous words}} = \frac{1148}{1500} = 76.53\%$$

The results are furnished in the following Table 1.

Table 1: Comparison of Results

Algorithm	**% of Accuracy**
First Sense TWSD	32.86%
SSTWSD with Word Embeddings	76.53%

CONCLUSION

Proposed method based on the idea of substituting the ambiguous word by any one of monosemous synonyms corresponding to the implied sense does not change the information conveyed by the sentence. Based on above observation, sense-specific embeddings of each ambiguous word are computed. Utilizing the word vectors obtained from training popular word embedding approach word2vec with the Telugu text corpus and sense-specific embeddings, a Semi-supervised WSD for Telugu is proposed. From the comparison experiment, it is shown that WSD coupled with word2vec architecture provides competent results. The advantage of this approach is that with the minimal data as input it is resulting in state-of-the art WSD system. It can be easily extended to other languages which are constrained by the availability of resources.

REFERENCES

[1]. G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. J. Miller, “WordNet: An on-line lexical database,” International Journal of Lexicography, Vol. 3, No. 4, pp. 235-244, 1990.

[2]. https://www1.essex.ac.uk/linguistics/external/clmt/w3c/corpus_ling/content/corpora/list/priv ate/brown/brown.html.

[3]. Gaizauskas, “Gold Standard Datasets for Evaluating Word Sense Disambiguation Programs. Computer Speech and Language, Vol. 12, No. 3, Special Issue on Evaluation of Speech and Language Technology, pp. 453-472, 2009.

[4]. G. Nagaraju, N. Mangathayaru, B. Padmajarani. “Transition based parser for Telugu language”. International Journal of Engineering and Technology, Vol.7, No.4, pp: 4674- 4677, 2018.

[5]. G. Nagaraju, N. Mangathayaru, B. Padmaja Rani. “Dependency Parser for Telugu language”. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, Article No. 138, 2016. https://doi.org/10.1145/2905055.2905354.

[6]. Pushpak Bhattacharyya, “IndoWordNet” Department of Computer Science and Engineering Indian Institute of Technology Bombay http://www.cfilt.iitb.ac.in/indowordnet/index.jsp.

[7]. M. Lesk, “Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone,” Proceedings of SIGDOC, 1986.

[8]. S. Banerjee and T. Pedersen, “An adapted Lesk algorithm for word sense disambiguation using WordNet,” In Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, February 2002.

[9]. Prity Bala, “Knowledge Based Approach For Word Sense Disambiguation Using Hindi Wordnet,” The International Journal Of Engineering And Science (IJES), Volume 2 Issue 4, PP. 36-41, 2013.

[10]. Alok Ranjan Pal, DigantaSaha, and Sudip Kumar Naskar, “ Word Sense Disambiguation in Bengali: a Knowledge based Approach using Bengali WordNet” 2017 IEEE .

[11]. G Nagaraju, N Mangathayaru, B Padmaja Rani. “MST Parser for Telugu Language”. Proceedings of the Third International Conference on Computational Intelligence and Informatics, pp: 271-279, 2020.

[12]. G. Nagaraju, N. Mangathayaru, B. Padmajarani “Integrating Transition and Graph Based Dependency Parsers Using Ensembled and Stacking Approaches for Parsing Telugu Language”, International Journal of Advanced Research in Engineering and Technology (IJARET), Vol.12, No.2, PP.96-105, Feb 2021.

[13]. Roberto NAVIGLI and Paola VELARDI, “Structural Semantic Interconnection: a knowledge-based approach to Word Sense Disambiguation. SENSEVAL-3: International International Workshop on the Evaluation of Systems for the Semantic Analysis of Text,” Barcelona, Spain, Association for Computational Linguistics, 2004.

[14]. E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In Proceedings of the 16th International Conference on Computational Linguistics, pages 16–22, Copenhagen, 1996.

[15]. 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 265–283. MIT Press, 1998.

[16]. D. Lin. Using syntactic dependency as a local context to resolve word sense ambiguity. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pages 64–71, Madrid, July 1997.

[17]. Patwardhan, S., Banerjee, S., Pedersen, T. (2003). Using Measures of Semantic Relatedness for Word Sense Disambiguation. In: Gelbukh, A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2003. Lecture Notes in Computer Science, vol 2588. Springer, Berlin, Heidelberg.

[18]. Singh, S., Singh, V. K. and Siddiqui, T. J.: Hindi Word Sense Disambiguation using Semantic Relatedness measure. In Proceedings of 7th Multi -Disciplinary workshop on Artificial Intelligence, Krabi, Thailand, pp. 247-256 (2013).

[19]. SuneethaEluru, VishalaSiddu, “A Knowledge Based Word Sense Disambiguation in Telugu Language”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-10 Issue-1, October 2020.

[20]. Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In Pro- ceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1059–1069, Doha, Qatar.

[21]. Reisinger, J. and Mooney, R. J. 2010. Multi-prototype vector- space models of word meaning. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 109–117, Los Angeles, CA, USA.

[22]. Huang, E. H., Socher, R., Manning, C. D., and Ng, 2012. Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1, ACL ’12, pages 873–882, Jeju Island, Korea.

[23]. Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781.

[24]. Banko, M., Brill, E., 2001. Scaling to very very large corpora for natural language disambiguation, in: Proceedings of the 39th annual meeting on association for computational linguistics, Association for Computational Linguistics. pp. 26–33.

[25]. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2013), Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119