ICSEE2019 International conference
JULY 2019 on systems energy
and environment

Deep Learning Models For Word Sense Disambiguation : A Comparative Study

Sandeep Nithyanandan
Dept of CSE (Computational Linguistics)
Govt. Engineering College,
Sreekrishnapuram, Kerala
sandeepn96@gmail.com

Raseek C
Dept. of CSE
Govt. Engineering College,
Sreekrishnapuram, Kerala
raseekc@gecskp.ac.in

***Abstract*—A specific word can have different meanings depending on the context in which it appears. Identifying the proper sense of the word is crucial in many tasks such as Machine Translation, Anaphora Resolution, Search Engine Recommendation. A word with wrong sense can make the whole sentence meaningless. The task of Word Sense Disambiguation (WSD), is to assign a sense to the ambiguous word based on the context. This is a basic problem which needs to be resolved in the field of NLP and has a variety of solutions. Usual approaches involve supervised machine learning techniques which uses bag-of-words approach. But to achieve better results, WSD should move to sequence modeling rather than the traditional bag of words approach for better performance. Deep neural networks have been used for variety of NLP tasks and a Bi-Directional Long Short-Term Memory (BiLSTM) or a Bi-Directional Gated Recurrent Unit (BiGRU) are possible candidate solutions. The proposed model uses one word per BiGRU or BiLSTM approach, where each word has a model trained for disambiguation. This helps in easy modification of existing model whenever needed. A comparative study on various deep learning models is also performed. The evaluation study of the models shows Bi-Directional models outperforming the other deep learning models.**

***Index Terms*—Word Sense Disambiguation, Bi-Directional Long Short-Term Memory, Bi-Directional Gated Recurrent Unit, GloVe word embeddings.**

I. INTRODUCTION

In the field of Computational Linguistics, word-sense disambiguation (WSD) is a problem that deals with identifying the correct sense of an ambiguous word in a sentence. The solution to this problem impacts other computer-related problems, such as discourse, improving relevance of search engines, anaphora resolution, coherence, inference.

In simple terms, words are generally ambiguous and depending on the context, it can have several related or unrelated meanings. For instance, the word ”bank” has senses ’river bank’ or ’a financial institution which lends money’. But when an ambiguous word comes in a particular context, the sense of the word gets resolved. The context in which the word appears determines the sense of the word. This case can be well understood from the following example: ’I went to the bank to deposit money’. The meaning of the word ’bank’ in the sentence is ’a financial institution which lends money’. The sense is disambiguated based on the context. This task of assigning a word token in a text, e.g. bank, to a properly defined meaning in a dictionary is called word sense disambiguation (WSD) .

There are variety of techniques from the basic dictionary based methods that relies on information from a lexical resources, to machine learning techniques that uses a trained classifier to disambiguate the sense of each word from a dataset, to unsupervised method that cluster occurrences of words, thereby disambiguating the sense.

Even though these techniques are effective , they suffer from two major flaws.

- The algorithms do not consider the word ordering which is essential to capture context.
- They rely heavily on language specific hand-crafted features for effective performance.

Deep learning has been apply on variety of Natural Language Processing(NLP) tasks and has proven to be very effective. WSD is still a task in which deep learning is yet to be fully explored. The paper considers different deep learning techniques which are used for resolving WSD. Deep learning can be effective because:

- It uses sequence processing and considers the word order which is essential for understanding the context.
- The deep learning models do not require any hand crafted features to be given as input since the features are learned automatically.

The paper focuses on different deep learning model for the task of Word Sense Disambiguation and a comparative study on which model performs the best is performed. The models that are being considered are the BiLSTM, BiGRU, Long Short-Term Memory(LSTM)[1], Gated Recurrent Unit(GRU). The paper focuses on deep neural network due to it’s widespread popularity for variety of Natural Language Processing tasks and it’s proven efficiency.

WSD is a classical NLP problem and the efficiency in sense disambiguation determines the performance of many other task such as Machine Translation, Information Retrieval etc.

This paper is organized as follows: Section 2 describes related works and Section 3 discusses about proposed system. Section 4 explained about evaluation and results of the system. Section 5 gives a brief concluding comments and about future directions.

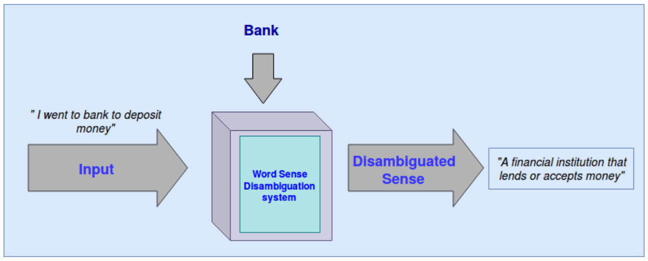
Bank
" I went to bank to deposit money"
Input
Word Sense Disambiguation system
Disambiguated Sense
"A financial institution that lends or accepts money"

Fig. 1. Basic Idea of Proposed System

II. RELATED WORKS

Word Sense Disambiguation(WSD) systems have used variety of approaches to perform the particular task. These include Dictionary based approaches, Supervised Learning techniques, Unsupervised Learning techniques and Deep Learning techniques.

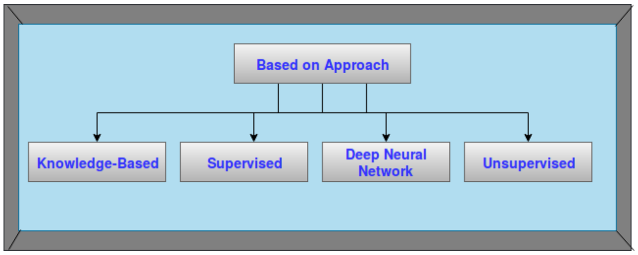

Fig. 2. Different approaches for WSD

M Postma et al. [2] in their work explores LSTM for the task of disambiguating the sense of an ambiguous word. Their approach is characterized by high performance, simplicity and the ability to extract a lot of information from raw text. They use SemCor and OMSTI dataset for the training. They perform the sense disambiguation in broadly 3 steps. (1) Use a large unannotated dataset to construct a language model. (2) Using a much smaller annotated data set to extract sense embedding from this model. (3)The predictions on the lemmas in unseen sentences are done using the sense embeddings.

A Pesaranghader et al. [3] proposes to use a Deep Bi-Directional LSTM to learn the context from the text. Deep Bi-Directional LSTM is used in-order to resolve the issue of one-word per LSTM approach having to create a single LSTM for each word. Here Deep networks of LSTM is created to consider senses and context sequences works on all ambiguous words collectively. This method effectively reduces the LSTMs needed to disambiguate. The additional number of Bi-Directional LSTM layers helps to essentially capture the context and resolve dependencies efficiently.

H Salomonsson et al. [4] in his work uses a Bi- Directional LSTM(BiLSTM) to identify the sense of the word in the sentence. The BiLSTM is shared across all the words. This helps the model in sharing statistical strength across the words, as a result, it scales well with the increase in vocabulary size. The model is trained from end-to-end, i.e. from raw data to the sense of the ambiguous words and it utilizes the word- order effectively. The word order has a huge importance in disambiguating the sense as the meaning of the word can change when the word order is changed. They use Glove embeddings to embed the words to vectors in high dimensional space. The use of high dimensional embeddings and effective use of word order increases the accuracy of the model exponentially.

M Okumura et al. [5] uses a Deep Belief Network(DBN) for the disambiugating the sense. DBN uses Restricted Boltzmann Machine (RBM) as a pre-training method to greedily train layer by layer. Then a separate fine tuning step is employed to improve the discriminative power. The system is evaluated on SEMEVAL-1 dataset.

S. Kwon et al. [6] in their work make use of knowledge based graph to disambiguate the sense. A new WSD technique is proposed by using similarities between an ambiguous word and words in the input document to generate the context of an ambiguous word Additionally, a new word similarity calculation method using BabelNet semantic network structure is performed.

S. Yamaki et al. [7] proposed a supervised learning model for Word Sense Disambiguation. They employ sentence similarities from context word embeddings to disambiguate the sense. If N example sentences exist in training data, the basic feature vector is added to the N-dimensional vector with N similarities between each pair of example sentences. This feature vector is used to train the classifier and disambiguate the sense. They use LinearSVC as the classifier algorithm.

F Hristea et al [8] in their work use an unsupervised algorithm for global word sense disambiguation inspired by DNA sequencing. The algorithm works in three main steps:

- A brute-force WSD algorithm is applied to short context windows selected from the document (up to 10 words) to generate a short list of likely sensory configurations for each window.
- The resulting local sense configurations are assembled on the basis of suffix and prefix matching in longer composite configurations.
- The resulting configurations are ranked by their length, and each word’s meaning is selected based on a voting scheme that only takes into account the top k configurations in which the word appears.

R Munot [9] uses Recurrent Neural Networks to learn the context embeddings. Using this context embedding the sense of the word is identified. One-word per LSTM model is employed. For every word to be disambiguated an LSTM and classifier is trained. The LSTM learns the context of the word from the training data and using a softmax activation function the classifier predicts the probability distribution of the senses of that particular word. One-word per LSTM model is easier to train and new words can be easily added to the model without needing to retrain the whole model.

TABLE I
COMPARISON OF VARIOUS METHODOLOGIES

Approach	Advantage	Disadvantage
Knowledge-Based	These algorithms give higher Precision	These algorithms are overlap based. So they suffer from overlap sparsity and performance depends on dictionary definitions.
Supervised	These algorithms are better than knowledge-based and unsupervised w.r.t. implementation perspective.	These algorithms don’ t give satisfactory result for resource scarce languages
Deep Neural Networks	These are the best algorithms which gives the maximum accuracy	These algorithms require large training data which may not be easily available.
Unsupervised	There is no need of any sense inventory and sense annotated corpora in this approach.	These algorithms are difficult to implement and performance is always inferior to supervised and deep neural networks.

III. PROPOSED METHODOLOGY

Figure 3 shows the basic architecture of proposed Word Sense Disambiguation system. The main three stages of the system architecture are dataset preprocessing, word embedding generation, training the Bi-Directional LSTM/GRU model for Sense Disambiguation. The Data Pre-processing stage performs the basic pre-processing of the dataset which makes it easier for computation. The Word Embedding stage performs a mapping of the words in the data to a vector space. This helps to resolve the context easily. The last stage uses a BiLSTM or BiGRU model for task of disambiguating the word.

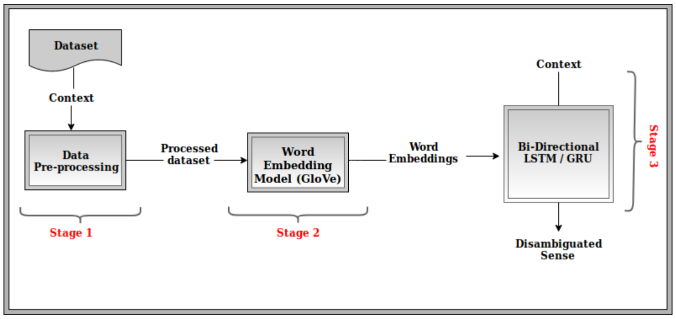

Fig. 3. Proposed System Architecture

A. Dataset Collection and Preprocessing

The dataset used is One-Million Sense Tagged Corpus (OMSTC) [10]. The dataset is divided into 4 categories: noun, adverb, verb, adjective. The dataset has many words, in which the context required is very less in number. The basic preprocessing involves removing those words which do not have context less than 1000. The dataset is in a format where the context and corresponding sense id of a word are in different files. The sense id is used to extract the sense of the word using Wordnet. The context is in an XML format. Both the context and sense are extracted and converted into a CSV file.

The input context is not directly suitable for training purpose. The input is pre-processed to remove the stop words, spell correction is performed to correct the wrong words. Spelling Correction stage is added in context pre-processing stage to remove any possible spelling mistakes that the user may make in the input query. Unwanted spaces, punctuation marks and stop words are also removed. Stop word removal is highly important as it removes words that don’t give much information to capture the context. Moreover, the stop words contributed to high computation time, removing the stop words helps to reduce the computational time.

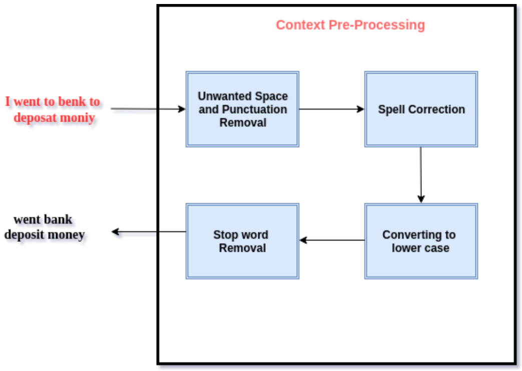

Fig. 4. Data Pre-Processing Task

B. Word Embedding Generation

Word embedding generation is the process where words are mapped to real-valued vectors. The neural networks accept only vectors as input. The inability to process raw text is the reason for converting the text into a suitable format for neural networks. There are several word embedding models like Word2Vec, skip-gram, CBOW, GloVe etc.

In the proposed system architecture, GloVe [11] model is used to generate the word embeddings. GloVe (Global Vectors) model makes use of global co-occurrence matrix for generating the word vector representations. The pre-trained GloVe vectors have shown promising results in a wide range of NLP applications. 200-dimensional pre-trained GloVe word vectors are used as the initial vectors. If a term is not present in GloVe vocabulary, then it will be initialized with the zero vector .

C. Building Bi-Directional LSTM/GRU Model

The embedded vectors are passed on to the deep learning model which uses either a Bi-Directional LSTM(BiLSTM) or a Bi-Directional GRU(BiGRU) . The model learns the various dependencies are disambiguates the sense of the ambiguous word. Bi-Directional model is preferred as it is able to see both past and future information, thereby resolving dependencies more. Bi-Directional model help to resolve more context for the problem solving and performs significantly more effective than uni-directional LSTMs or GRUs.

The features learned from the Bi-Directional LSTM/GRU is the used to disambiguate the sense. Finally, the feature representation is fed to the classification layer which uses Softmax function for disambiguating the sense. Cross-entropy is minimized while training the model and the optimizer used is Adam. To avoid over fitting dropout is added to the output of BiLSTM or BiGRU layer.

TABLE II
MODEL SPECIFICATION

Parameters	Value
Layers	Embedding, BiLSTM/BiGRU, Dropout, Flatten,Dense
Dropout	0.2
Activation	Softmax
Epochs	15
Batch Size	16

The proposed deep learning model is implemented using Keras [12] with Tensorflow [13]. The dropout value of 0.2 is found out to be the best value for the system. Softmax activation function is used as it gives a probability distribution among all the output classes. The *argmax* function is used to find the class with highest probability. This class will be the predicted output class of the input. The model is run for 15 epochs each for all the words in the model. Batch size of 16 is preferred because inorder to compensate the lack of training data, the model has to learn step by step seeing small amount of examples.

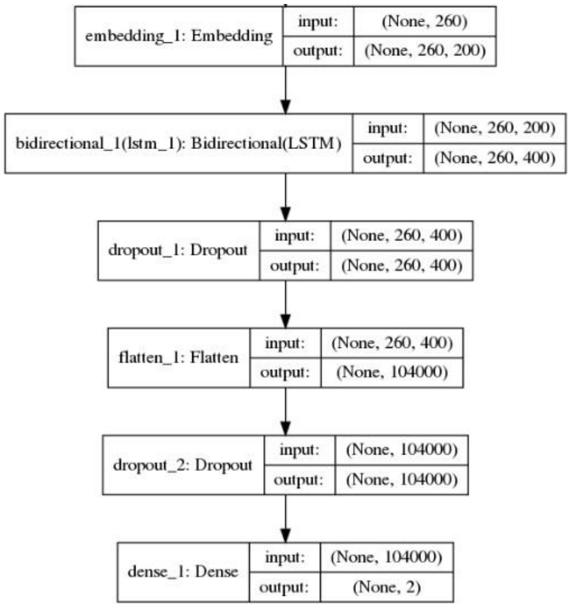

Fig. 5. Layers in BiLSTM

The training process is done for each word to be disambiguated in the system, thereby creating a one word per BiLSTM/BiGRU model. This method is highly efficient as it does not create memory related issues. One word per model is preferred as there need to retrain the whole model when a new word is to be added to the system.

IV. RESULTS AND DISCUSSION

The Bi-Directional models are compared with the Uni-Directional LSTM and GRU models. The performance metrics used to evaluate the model is accuracy. The testing accuracy of the Bi-Directional LSTM is 93% and the Bi-Directional GRU is 90%. Comparing with LSTM which has an accuracy of 89%, the Bi-Directional model is performing better than uni-directional LSTM.

Table III

COMPARISON OF PERFORMANCE OF VARIOUS MODELS

Model	Accuracy	Precision	Recall	F-Measure
BiLSTM + GloVe	93%	.90	.88	.889
BiGRU + GloVe	90%	.88	.87	.874
LSTM + GloVe	89%	.83	.85	.839
GRU + GloVe	85%	.81	.86	.834

Table II shows the performance of the models tested for the proposed WSD system. It can be clearly seen that the BiLSTM model performs better than rest of the models . A comparative study on WSD involving BiLSTM and BiGRU is never performed, so this work provides an insight on how the models work for the WSD task. It can be easily seen that since BiLSTM has more gates than BiGRU, the BiLSTM model is able to learn more context than the BiGRU model. This results in the increased accuracy of the BiLSTM model. The Bi-Directional model is better than Uni-Directional model because of the ability to process information from both direction. The two more insights that can be gathered from this study is that :

- The BiLSTM model takes more training time that Bi-GRU.
- BiLSTM has more accuracy than BiGRU.

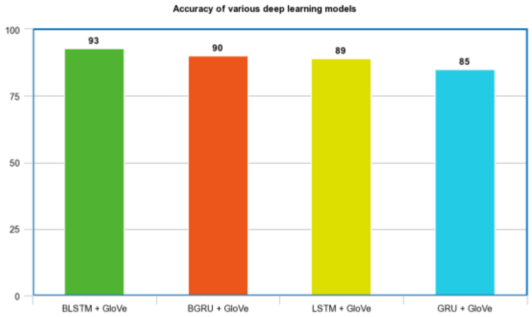

Fig. 6. Comparison Graph of Accuracy of various models

V.CONCLUSION AND FUTURE WORK

Words are ambiguous in nature and can have a variety of meanings. Depending on the context in which the word appears, the sense of the word can be disambiguated. This is a famous Natural Language Processing problem and has variety of solutions. Deep learning solution have provided efficient solution for variety of problems and is being used to solve this task.

The proposed system makes a comparative study on how BiLSTM and BiGRU works on Word Sense Disambiguation Task. Both model takes GloVe embeddings as their input and learn the context to predict the sense of ambiguous word. The findings from the comparative study is BiLSTM works better than BiGRU to predict the sense. Moreover one word per BiLSTM/BiGRU is proposed. This model is highly useful as it requires lesser memory to train and new words can be added easily without need to retrain the model.

The possible extensions can be:

- A proper and improved dataset can be created making the task of disambiguation easier and efficient.
- A stacking of BiLSTM can be used to extract more features from the context, thereby introducing a new methodology which can be an extension to this work.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] M. Le, M. Postma, J. Urbani, and P. Vossen, “A deep dive into word sense disambiguation with lstm,” in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 354– 365.

[3] A. Pesaranghader, A. Pesaranghader, S. Matwin, and M. Sokolova, “One single deep bidirectional lstm network for word sense disambiguation of text data,” in *Advances in Artificial Intelligence: 31st Canadian Conference on Artificial Intelligence, Canadian AI 2018, Toronto, ON, Canada, May 8–11, 2018, Proceedings 31*. Springer, 2018, pp. 96–107.

[4] M. Kågebäck and H. Salomonsson, “Word sense disambiguation using a bidirectional lstm,” *arXiv preprint arXiv:1606.03568,* 2016.

[5] P. Wiriyathammabhum, B. Kijsirikul, H. Takamura, and M. Okumura, “Applying deep belief networks to word sense disambiguation,” *arXiv preprint arXiv:1207.0396*, 2012.

[6] O. Dongsuk, S. Kwon, K. Kim, and Y. Ko, “Word sense disambiguation based on word similarity calculation using word vector representation from a knowledge-based graph,” in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2704–2714.

[7] S. Yamaki, H. Shinnou, K. Komiya, and M. Sasaki, “Supervised word sense disambiguation with sentences similarities from context word embeddings,” in *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Oral Papers*, 2016, pp. 115– 121.

[8] A. M. Butnaru, R. T. Ionescu, and F. Hristea, “Shotgunwsd: An unsupervised algorithm for global word sense disambiguation inspired by dna sequencing,” *arXiv preprint arXiv:1707.08084*, 2017.

[9] R. Munot, “Word sense disambiguation using rnns for context embed- ding,” 2015.

[10] K. Taghipour and H. T. Ng, “One million sense-tagged instances for word sense disambiguation and induction,” in *Proceedings of the nine- teenth conference on computational natural language learning*, 2015, pp. 338–344.

[11] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[12] F. Chollet *et al.*, “Keras,” 2015.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/