IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 6, June 2020

Word Sense Disambiguation (WSD) using Neural Networks

Ms.Nilima .S.Chaudhari[1]

PG Student, Dept. of Computer Engineering., Matoshri College of Engineering[1]

Abstract: Word Sense Disambiguation (WSD) is the task of removing ambiguity in different senses of words. It is a core research field in computational linguistics dealing with the auto- matic assignment of senses to words occurring in a given context [11]. Humans are inherently good at WSD and distinguish senses used in words through spoken language. Computers on the other hand have difficulties identifying correct senses of words. Various advancements have been made in the task of disambiguation using mainly four approaches: Knowledge-based, Supervised, Semi-Supervised, and Unsupervised. Better understanding of the human language will help computer's performance in various applications such as search engine optimization, information retrieval, information extraction, software assistants, and voice command interpretation. The objective of this work is to present a supervised neural network machine learning model using various algorithms dedicated to the task of maximizing accuracy of sense detection. The input layer of the neural network will consist of nodes having binary values depending on the presence or absence of frequently occurring context words related to the ambiguous words. The output layer will consist of nodes equal to the number of senses the ambiguous word has. Training and testing of the model will be done using lexical resources such as SemCor or OMSTI. Accuracy will be calculated based on All- Word tasks from SemEval International Workshops

Keywords: Machine Learning,Neural Network, Classifica- tion Algorithm

I. INTRODUCTION

Word-Sense Disambiguation (WSD) is a branch of Natural Language Processing (NLP) which specifies some open prob- lems concerned with identifying the correct sense of a word used in a respective sentence. Many words used in the English language have various different senses or meanings. WSD is concerned with the problem of selecting the correct meaning. The solution to this problem impacts improving relevance of search engines.The human mind is very proficient at word-sense disam- biguation. Simple context is all that is needed for humans to understand the correct sense or meaning of a word. Human languages have developed due to the intellectual ability of neural networks in human brain. In computer science it has been a long-term challenge to develop the ability in computers to perform language processing on the scale that humans do. For example, consider a word bass in English which has two meanings: any of various North American lean-fleshed freshwater fishes and the other: denoting the member of a family of instruments that is the lowest in pitch.

A. Introduction to Machine Learning

Machine Learning (ML) is defined as programming comput- ers to optimize a performance criterion using example data or past experience. In our system, the performance criterion is the accuracy of the model on testing data. Supervised ML consists of 2 main parts: Training and Testing. Training comprises of feeding labelled data into the model to gain experience. Testing comprises of predicting outputs by trained model based on experience.

B. Machine Learning in WSD

As stated before, the human brain is masterful at distin- guishing between various senses of a word based on their context. The best way to reproduce this capability within machines is to make the computer think like humans do, allow it to learn from experience and make predictions based on this experience. The way this is implemented is Machine Learning, specifically using a Neural Network.

C. Artificial Neural Networks

A neural network is a network or circuit of neurons composed of artificial neurons or nodes. Artificial Neural Networks (ANN) are composite layers of compute units that process data individually in order to simulate the working of a human brain. Similar to human brains, ANNs learn with experience and show improvements in tasks when data available is increased. ANN consisting of one or two hidden layers are called shallow neural networks and those with more hidden layers are called deep neural networks.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 6, June 2020

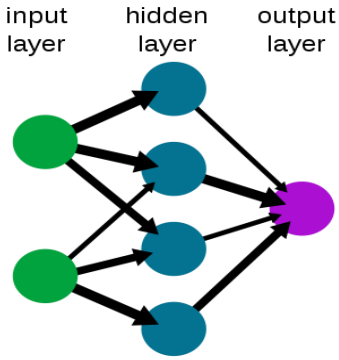

Fig. 1. Simple Neural Network

II. RELATED WORK

A large number of methods of Word Sense Disambiguation (WSD) have been studied and researched in the past. These methods mostly include four different approaches to WSD: Knowledge based, Supervised, Semi-Supervised, and Unsupervised.

A. Knowledge-based Approach

Knowledge based algorithms use various lexical resources such as Machine-Readable Dictionaries (MRDs), WordNet to identify the correct sense of words.
These Algorithms are easy to implement and were the first to be developed while trying to solve the problem of WSD. A knowledge-based system only needs access to commercial dictionary resources to start process of disambiguation.
Drawback of these algorithms is that their performance is limited on the speed of searching and retrieval of these resources. As the size of the resources increase, so does the latency and hence performance decreases. [1]

B. Supervised Approach

Supervised methods are called so because they require human assistance. Large amount of labelled data is required to make supervised models perform as expected. The larger the data set available, the greater is the prediction accuracy of these system.
A learning set is prepared for the system to predict the actual meaning of an ambiguous word using a few sentences, having a specific meaning for that particular word. A system finds the actual sense of an ambiguous word for a particular context based on that defined learning set [1].
Supervised approach always gives superior performance than any other methods. However, these supervised methods require large data sets, and are therefore limited in their capabilities. Such data requires manual tagging of senses which is an expensive and time-consuming task.

C. Semi-supervised Approach

Many word sense disambiguation algorithms use semi- supervised learning which provides a sort of compromise between supervised and unsupervised approaches. They allow both labelled and unlabelled data and are therefore useful when there is a lack of training data. The bootstrapping method starts from a small amount of seed data for each word: either a small number of sure-fire decision rules (e.g., 'play' in the context of 'bass' almost always states the musical instrument) or manually tagged training corpus.
Using any of the supervised methods, small amount of tagged or labelled data is used to train an initial classifier. This classifier is then fed unlaballed data in order to extract a larger labelled dataset in which only the perfect classifications are included. Such processes are usually iterative each iteration training being done on a successively larger dataset. The obtained data set becomes larger and larger until we stop the process after a certain number of iterations have been reached or the maximum size of dataset is reached.

D. Unsupervised approach

Unsupervised learning methods are the most difficult to implement for WSD researchers. Using Unsupervised approaches, we basically mean to say that word senses can be deduced using other similar sentences. Using Clustering algorithms, such sentences with a certain degree of similarity can be grouped together with each cluster specifying one sense of a word. This process is called Word Sense Induction.
As expected, the performance of such algorithms has been shown to be less than the other methods of WSD due to lack of training data but it is hoped that in the future, the unsupervised techniques can successfully overcome the problem of scarcity of expensive manually tagged data in order to be the most efficient sense prediction approach.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 6, June 2020

III. PROPOSED ALGORITHM

Various solutions to word sense ambiguity have been put forward. Most of these systems use one of the four approaches mentioned earlier. Out of these approaches, supervised ap- proach to WSD has been proven to produce maximum accuracy. Therefore, in our proposed model we will be making use of supervised approach as well. As mentioned earlier, Artificial Neural Networks mimic the in which an ambiguous word occurs, the neural net should be able to successfully predict the correct sense of the ambiguous word. For creating a accurate neural net classifier we also require large amounts of labelled data as such a model falls under the supervised approach to WSD. We will be using SemCor [7] and OMSTI [8] labelled data sets for this purpose. Once trained, we can judge the accuracy of the Neural Net by using the test data set. The classifier should also be able to return the correct sense of an ambiguous word based on input given by user.

A. Word Embeddings

The input feature vector of an ambiguous word for the neural network will be created using its word embeddings. The vectors we use to represent words are called neural word embeddings. Word Embeddings are created using the words similar and most commonly used context words. They can be created using various methods such as Word2Vec, Recurrent RNN models such as LSTM, etc. Word Embeddings measure cosine similarity,i.e. no similarity is expressed as a 0, while total similarity is expressed as 1.

Example:

Figure 2 shows word embeddings for the word 'Sweden'. Since Norway and other Scandinavian countries are closely related to Sweden, their cosine values are closest to 1.

B. Feature Vectors

The input feature vectors to the Neural Network will be created using both the input of dataset and the word embeddings of the ambiguous word. If embedding words are present in the input context then they will be represented in the feature vector with value '1', else with a '0'. After every word in input context is checked for its presence in word embeddings, the resultant feature vector will be passed to the Neural Net input layer. Therefore, number of nodes in the input layer of Neural Net will be equal to the number of word embeddings we use.

Example:

Assume that the word embeddings for the word 'crown' are: [jewels king teeth drill dentist]

and that input is the sentence: "The dentist did a really good job putting the crown on my teeth" Then the Input Feature vector will be:

[0 0 1 0 1]

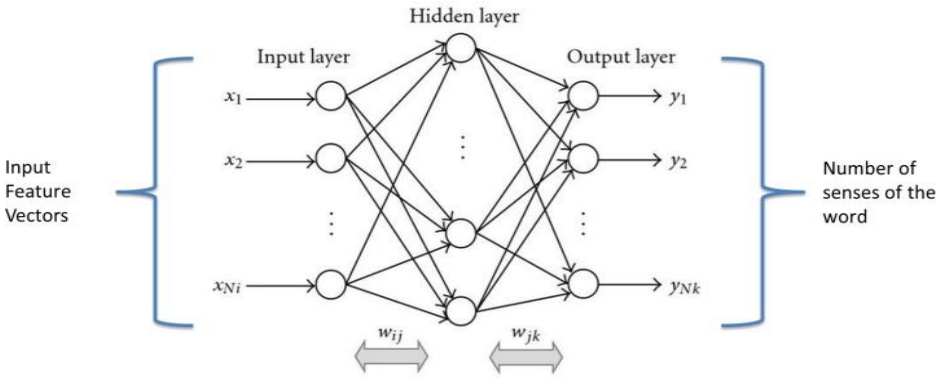

Fig. 3. Neural Network for an Ambiguous Word

A. Feed Forward Neural Networks

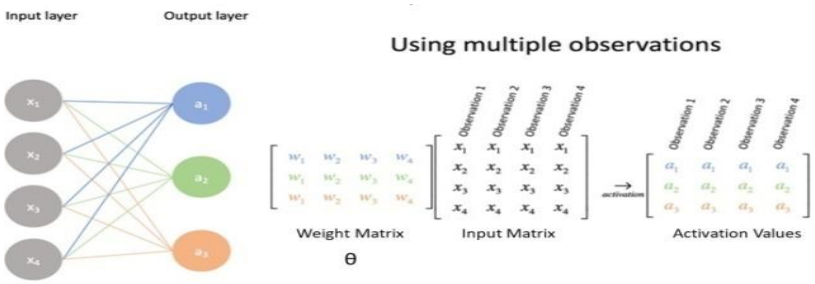

Fig. 4. One step of forward propagation

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 6, June 2020

The following figure shows an example of a forward prop- agation step in a Feed Forward Neural Network. In vectorised implementations, input sets are represented in columns of a matrix which are multiplied by a weight matrix theta that represents each layer of a neural network. The use of matrices reduces time and increases efficiency for calculations as it does not require loops in the program structure to multiply each element individually.

B. Cost Calculations of Neural Network

Calculating costs is the one definitive way of understanding that our Neural Network is working correctly. After every iteration the cost of the neural network is calculated using the cost function given below. Displaying the value of cost every few hundred iterations can help us accurately gauge whether our neural net is actually learning or not.

$$h_\Theta(x) \in \mathbb{R}^K \quad (h_\Theta(x))_i = i^{th} \text{ output}$$

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log(h_\Theta(x^{(i)}))_k + (1 - y_k^{(i)}) \log(1 - (h_\Theta(x^{(i)}))_k)\right]$$
$$+\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(\Theta_{ji}^{(l)})^2$$

Where:

- m = number of features
- K = number of output layer nodes
- L = number of layers in network
- θ represents weight matrix

Fig. 5. Cost Function for Classification Algorithms

C. Output of Neural Network

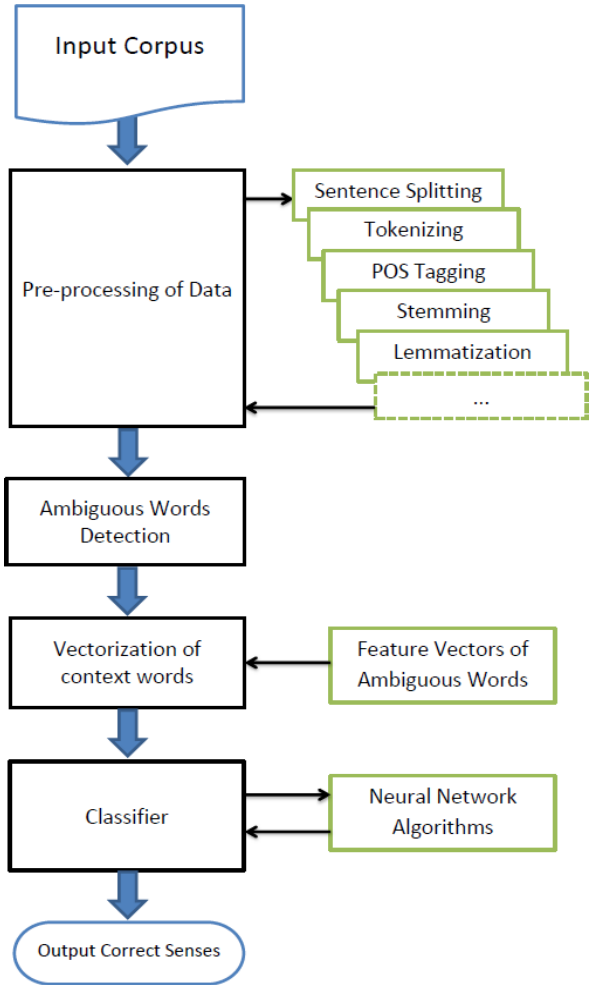

Fig. 6. System Architecture

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 6, June 2020

The output layer of neural network contains nodes equal to the number of different senses for the ambiguous word, according to the WordNet [6] dictionary. The node for which the highest numerical value is calculated among the different output nodes will represent the predicted sense. If the third node of output layer has the highest value, then it means that the system has predicted the third sense, all senses being annotated by WordNet [6].

IV. SIMULATION RESULTS

Implementation Status

The entire implementation was completed as per schedule with little to no delays. The most the project deviated from schedule was during the data processing stage. The extreme computing requirement of processing the 1.5 gigabyte dataset of OMSTI was proving extremely hard to work with. The computing environment available to us was simply not su cient for effciently fnishing work in a timely manner. The team then made a decision to transfer the data processing module to the Google Colabolatory where we could get the computing power of Google servers with 25 Gigabytes of RAM at our disposal. For this to work we rst had to upload the said OMSTI dataset to Google Cloud for working on it. After we got the module to run, the rest of the project got back on schedule and was completed with ease. As of now, the Neural Network training and testing framework is completed as well as the data processing unit for creation of data inputs to neural net.

Testing Strategy

The testing of the project, was done in a V model basis. Right after realizing the business requirements of the system, test cases for acceptance testing were being developed. The modular structure of our core developement meant it allowed us to begin testing at a very early stage. Initial modules were unit tested using sin- gular inputs from the OMSTI dataset. After several modules were developed, their integration with the overall system was tested using component testing through comprehensive testing data extracted from OMSTI and SemCor.

Upon development of the modules, the system integration testing was started. Various erroneous inputs were given to the models as a way to test rhobustness of the system. Once system was su ciently tested, the accuracy calculations were begun.

After user interface was designed and implemented, its testing was con- sequently begun. GUI error messages were added to handle incorrect inputs and testing purposes.

Accuracy Results

The Neural Network framework created was tested for accuracy calculated using the Keras library. The sense-tagged dataset obtained from OMSTI is divided into training and testing data based on a exible ratio. The testing was done on a word by word basis, using the keras function, predict(). Giving the testing data as input to the function, it returns accuracy of the model.

For testing purposes, we considered the model of 76 di erent words that are generally considered to be ambiguous. The average training accuracy of these models came out to be 79.322%. These results ranged from 70% to 88% generally dependent on the data input size that was used to train the model. The testing accuracy was initially extremely low i.e. around 20%. But after regularization and adjusting hyperparameters, we were able to raise the testing accuracy upto 67% on previously unseen data. Based on previous research work conducted in this eld, we can say with su cient evidence that our results are almost upto the mark of industry leading algorithms.

System	Classi er	Training Data	Accuracy Average
Yuan and Team[14]	LSTM LSTM	SemCor OMSTI	0.834 0.799
Our Proposed Model	ANN ANN	SemCor OMSTI	0.79322 0.67

V. CONCLUSION AND FUTURE WORK

This work proposed a Word Sense Disambiguation (WSD) Model using Neural Network Algorithms that aim to maxi- mize accuracy for the given natural language processing task. Building on previous research work, our system hopes to further improve sense prediction accuracy and help in human- computer interfacing for future applications.

REFERENCES

[1]. Chandrakant D. Kokane, Sachin D. Babar "Supervised Word Sense Dis- ambiguation with Recurrent Neural Network Model" 2019 International Journal of Engineering and Advanced Technology (IJEAT)
[2]. Pratibha Rani, Vikram Pudi, Dipti M. Sharma "Semi-supervised Data-Driven Word Sense Disambiguation for Resource- poor Languages"
[3]. 2017 14th International Conference on Natural Language Processing (ICON)
[4]. Ignacio Iacobacci, Mohammad Taher Pilehvar, Roberto Navigli "Embedding for Word Sense Disambiguation: An Evaluation Study" 2016 Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 897-907
[5]. Udaya Raj Dhungana, Subarna Shakya, Kabita Baral and Bharat Sharma "Word Sense Disambiguation using WSD Specific WordNet of Polysemy Words" 2015 Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing
[6]. Michael Lesk// Automatic Sense Disambiguation using Machine Read- able Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone// Bell communications Research Morristown,NJ 07960
[7]. George A. Miller WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41 Bell communications Research Morristown,NJ 07960 https://wordnet.princeton.edu/
[8]. George A. Miller SemCor: Semantically Annotated English Corpus Princeton University
[9]. Kaveh Taghipour and Hwee Tou Ng One Million Sense-Tagged Instances for Word Sense Disambiguation and Induction 2015 Conference on Computational Language Learning
[10]. https://muse.dillfrog.com/ Online Dictionary
[11]. Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, Eric Al- tendorf 2016 Semi-Supervised Word Sense Disambiguation with Neural Models Google, Mountain View CA, USA
[12]. Mohammad Taher Pilehvar, Roberto Navigli. A large-scale pseudo word- based evaluation framework for state-of-the-art word sense disambigua- tion 2014