ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 1, Issue 5, June 2016

WORD SENSE DISAMBIGUATION USING CONTEXT CLUSTERING

[1] Pelja Paul.N, [2] Binu R, [3] Dibin Joseph
[1]M.Tech student, Computational Linguistics Govt. Engineering College, Palakkad, India
[2]Professor, Computer Science ,Govt. Engineering College, Palakkad, India
[3]Research Engineer , Cognicor Technologies Pvt Ltd, Infopark,kochi
[1] peljapaul@gmail.com , [2] binurajappan@yahoo.com , [3] dibin@cognicor.com

Abstract: ***Automatic multi class text classification is a machine learning task which categorizes document to one among a predefined set of classes. In recent years, deep learning technique such as Recurrent Neural Networks (RNNs) has become state-of-the-art model for a variety of machine learning problems. This paper introduces the scope of Long Short Term Memory (LSTM) - a type of RNN, for multi class text classification. LSTMs are capable of learning long-term dependencies while avoiding the vanishing gradient problem usually found in neural network algorithms. The proposed system is carried out in Reuters corpus, a dataset of 11,228 news wires from Reuters, labeled over 46 topics.***

Keywords: Context cluster; Tf-idf Vectorizer; Unsupervised; Machine translation; Word sense disambiguation.

I. INTRODUCTION

WSD is identifying the sense of a word used in a sentence when the word has multiple meanings. Three types of WSD approaches are present, Knowledge-based (dictionary), Supervised and Unsupervised methods. The Lesk method is the dictionary-based method. It is based on that words used together in the text are related to each other and that the relation can be observed in the definitions of the words and their senses. Two (or more) words are disambiguated by finding the pair of dictionary senses with the greatest word overlap in their dictionary definitions. Supervised methods are based on the context can provide enough evidence on its own to disambiguate words . supervised methods are subject to a new knowledge acquisition bottleneck since they rely on substantial amounts of manual sense tagged corpora for training, which are laborious and expensive to create. The unsupervised learning underlying assumption is that similar senses occur in similar contexts, and thus senses can be induced from the text by clustering word occurrences using some measure of similarity of context. New occurrences of the word can be classified into the closest induced clusters/senses.

Word Sense Disambiguation (WSD) consists of selecting the appropriate sense for a particular contextual occurrence of a polysemous word. This is related with the sense definitions. For instance, word sense induction refers to the process of discovering different senses of an ambiguous word without prior information. Word sense disambiguation is the task of assigning a meaning to an ambiguous word given the context in which it occurs. WSD serves as an intermediate step for many computer science applications such as machine translation, information retrieval, hypertext navigation, content and thematic analysis, speech processing. It has been a central problem since the earliest days of computational studies of natural language. Supervised learning requires many training sentences for each word. Bearing in mind that even in English, for which the most extensive research has been carried out historically, the sense tagged corpora are rather limited. It is a crying necessity to make better use of untagged corpora to be able to perform word sense disambiguation for any word in a running text [6]. Clustering methods have been extensively used in many Information Processing tasks in order to capture unknown object categories. Clustering has been scarcely used as a sense labeling method for Word Sense Disambiguation, as a way to identify groups of semantically related word senses that can be successfully used in a disambiguation process.

II. RELATED WORKS

A. Word Sense Disambiguation: a Survey

Conducted a survey on WSD in different international and Indian languages. Different approaches adopted from different research works. Indian languages have the large scale of morphological inflections. The context Clustering method is based

on clustering techniques. Context vectors are created and then they will be grouped into clusters to identify the meaning of the word. Vector space used as word space and its dimensions are words. A word in a corpus will be denoted as vector and how many times it occurs will be counted within the context . A co-occurrence matrix is created and similarity measures are applied. Discrimination is performed using any clustering technique [1].
WSD can be applied on machine translation (MT), information retrieval (IR) , information extraction (IE) and text mining. MT is used in WSD, a few words in every language have different translations based on the contexts of their use. Information retrieval (IR) used for resolving ambiguity in a query. And finding the exact sense of an ambiguous word. Information extraction is used in different research works as Bioinformatics research, Named Entity recognition system, co-reference resolution etc.

B. Context Clustering for Word Sense Disambiguation Based on Modeling Pairwise Context Similarities

Word sense disambiguation (WSD) gives a different context model for each individual word. The correlation regularity between the sense distinction and the context distinction can be captured at the category level, independent of individual words. A maximum entropy model is used for context clustering. It is trained to represent the generative probability distribution of context similarities based on heterogeneous features. Statistical annealing is used to derive the final context clusters [2].

To compute the context similarities, each context contains the following two categories of features:

i) Trigger words centering around the key word within a predefined window size equal to 50 tokens to both sides of the key word.
ii) Parsing relationships associated with the keyword automatically decoded by parser.
Based on the above context features, the following categories of context similarity features are defined:

(1) Context similarity based on a vector space model using co-occurring trigger words: the trigger words centering around the key word are represented as a vector, and the tf*idf scheme is used to weigh each trigger word. The cosine of the angle between two resulting vectors is used as a context similarity measure [5].

(2) Context similarity based on Latent semantic analysis (LSA) using trigger words: LSA is a technique used to uncover the underlying semantics based on co-occurrence data. Using LSA, each word is represented as a vector in the semantic space. The trigger words are represented as a vector summation. Then the cosine of the angle between the two resulting vector summations is computed, and used as a context similarity measure.

C. Sense Clusters

Sense Clusters creates clusters which made up of the contexts of given target word occurs. All the instances in a cluster are contextually similar to each other. The given target word has been used with the same meaning in all of instances. Each instance have 2 or 3 sentences, one of the sentence contains the given occurrence of the target word. Sense Clusters was intended to discriminate among word senses. Sense Clusters used for applications such as email sorting and automatic ontology construction. Based on a set of features that are identified from raw corpora Sense Clusters distinguishes among the different contexts in which a target word occurs [3].

Sense Clusters uses the N gram Statistics which is able to extract surface lexical features from large corpora using frequency cutoffs and various measures of association, including the log–likelihood ratio, Pearson's Chi–Squared test, Fisher's Exact test, the Dice Coefficient, Point wise Mutual Information, etc. Sense Clusters allows for the selection of lexical features from corpus of training data, from the same data that is to be clustered, which we refer to as the test data. Selecting features from separate training data is particularly useful when the amount of the test data to be clustered is too small to identify interesting features. Once features are selected, Sense Clusters creates a vector for each test instance to be discriminated where each selected feature is represented by an entry/index. Each vector shows if the feature represented by the corresponding index occurs or not in the context of the instance , or how often the feature occurs in the context.

Sense clustering is that meaningful word senses must be associated by means of a certain complex relation. To identify cohesive groups of senses which are assumed to represent different meanings for the set of words W. Those clusters that fit in with the context T contain the suitable senses. Disambiguate a set of related words at once using a given textual context.

Using a sense representation that overcomes the sparseness of WordNet relations, and that relates semantically word senses. Topic signatures built from WordNet and the Extended Star clustering algorithm. The way this clustering algorithm relates sense representations resembles the manner in which syntactic or discourse relations link textual components [5].

D. Scikit-learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Unsupervised learning, in which the training data consists of a set of input vectors x without any corresponding target values. The goal in such problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high dimensional space down to two or three dimensions for the purpose of visualization. In a large text corpus, some words will be very present (e.g. “the”, “a”, “is” in English) hence carrying very little meaningful information about the actual contents of the document. If we were to feed the direct count data directly to a classifier those very frequent terms would shadow the frequencies of rarer yet more interesting terms. In order to re-weight the count features into floating point values suitable for usage by a classifier it is very common to use the tf–idf transform. Tf means term- frequency while tf–idf means term-frequency times inverse document-frequency. This was originally a term weighting scheme developed for information retrieval (as a ranking function for search engines results), that has also found good use in document classification and clustering [4].

Clustering of unlabeled data can be performed with the module sklearn cluster. Clustering algorithm comes in two variants, a class, that implements the fit method to learn the clusters on train data, and a function, that, given train data, returns an array of integer labels corresponding to the different clusters. For the class, the labels over the training data can be found in the labels attribute. MeanShift and KMeans take data matrices of shape [n_samples, n_features]. These can be obtained from the classes in the sklearn feature_extraction module. MeanShift and Kmeans work with points in a vector space.

The KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields. The k-means algorithm divides a set of N samples X into K disjoint clusters C, each described by the mean of the samples in the cluster. The means are commonly called the cluster centroids. The K-means algorithm aims to choose centroids that minimize the inertia. Inertia makes the assumption that clusters are convex and isotropic, which is not always the case. It responds poorly to elongated clusters, or manifolds with irregular shapes. Inertia is not a normalized metric: we just know that lower values are better and zero is optimal. But in very high-dimensional spaces, Euclidean distances tend to become inflated. Running a dimensionality reduction algorithm such as PCA prior to k-means clustering can alleviate this problem and speed up the computations.

The MiniBatchKMeans is a variant of the KMeans algorithm which uses mini-batches to reduce the computation time, while still attempting to optimise the same objective function. Mini batches are subsets of the input data, randomly sampled in each training iteration. These mini-batches drastically reduce the amount of computation required to converge to a local solution. In contrast to other algorithms that reduce the convergence time of k- means, mini-batch k-means produces results that are generally only slightly worse than the standard algorithm. The algorithm iterates between two major steps, similar to vanilla k-means. In the first step, b samples are drawn randomly from the dataset, to form a mini-batch. These are then assigned to the nearest centroid. In the second step, the centroids are updated. In contrast to k- means, this is done on a per-sample basis. For each sample in the mini-batch, the assigned centroid is updated by taking the streaming average of the sample and all previous samples assigned to that centroid. This has the effect of decreasing the rate of change for a centroid over time. These steps are performed until convergence or a predetermined number of iterations is reached. MiniBatchKMeans converges faster than KMeans, but the quality of the results is reduced.

III. PROPOSED SYSTEM

WSD related contexts are grouped into different folders. Convert these contexts into context vectors. Tf-idf Vectorizer is used for the vector conversion. These vectors used for the creation of clusters. Tf-idf Vectorizer takes enough features for cluster formation. Appropriate clusters formed by k_means clustering using those features. Each cluster corresponds to one sense. In

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 1, Issue 5, June 2016

K_means clustering, a random centroid is selected. Then iteratively add vectors to that cluster. A new context comes to the system it added to appropriate cluster for WSD purpose. After that predict, it added to which cluster. For this prediction, we needed to convert the new context into

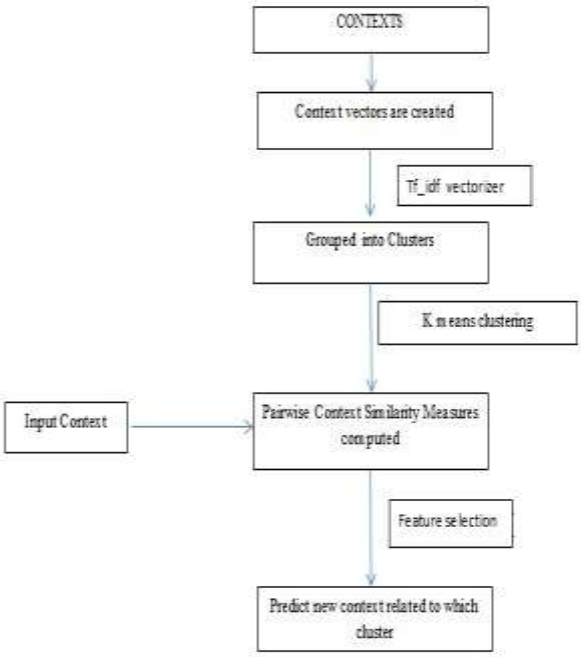

Fig1:Proposed System.

vectors. And also Tf-idf Vectorizer used for vector conversion and feature selection. When a number of features for clusters and new context become equal prediction take place. As shown in the Fig 1.

Non-adjusted measures like V-Measure, show a dependency
between the number of clusters and the number of samples. The mean V-Measure of random labeling increases significantly as the number of clusters is closer to the total number of samples used to compute the measure. Adjusted for chance measure such as ARI display some random variations centered around a mean score of 0.0 for any number of samples and clusters. Only adjusted measures can hence safely be used as a consensus index to evaluate the average stability of clustering algorithms for a given value of k on various overlapping sub-samples of the dataset.

Evaluating the performance of a clustering define separations of the data similar to some ground truth set of classes or satisfying some assumption such that members belong to the same class are more similar that members of different classes according to some similarity metric. Adjusted Rand index is a function that measures the similarity of the two assignments, ignoring permutations and with chance normalization. Adjusted rand score is symmetric. Swapping the argument does not change the score. It can thus be used as a consensus measure. Random (uniform) label assignments have a ARI score close to 0.0 for any value of n_clusters and n_samples . Bounded range [-1, 1]: negative values are bad (independent labelings), similar clusterings have a positive ARI, 1.0 is the perfect match score.

The Mutual Information is a function that measures the agreement of the two assignments, ignoring permutations. Two different normalized versions of this measure are available, Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI). NMI is often used in the literature while AMI was proposed more recently and is normalized against chance. Bounded range [0, 1]: Values close to zero indicate two label assignments that are largely independent, while values close to one indicate significant agreement. Values of exactly 0 indicate purely independent label assignments and a AMI of exactly 1 indicates that the two label assignments are equal (with or without permutation).

Homogeneity defines each cluster contains only members of a
single class. Completeness defines all members of a given class are assigned to the same cluster. The V-measure is actually equivalent to the mutual information (NMI). V_measure_score is symmetric: it can be used to evaluate the agreement of two independent assignments on the same dataset. The Silhouette Coefficient is defined for each sample and is composed of two scores:
a: The mean distance between a sample and all other points in the same class.
b: The mean distance between a sample and all other points in the next nearest cluster.

Higher Silhouette Coefficient score relates to a model with better defined clusters. The score is bounded between -1 for incorrect clustering and +1 for highly dense clustering. Scores around zero indicate overlapping clusters. The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster.

IV. OBSERVATIONS

Tf-idf vectorizer used for feature selection. It is used for cluster making. By an increasing number of features, improve the cluster quality. Then prediction became more accurate. System evaluated with 500 documents. From the results system has approx.86% accuracy. The system selects 1678 features for cluster creation. Clustering accuracy evaluated using Homogeneity, Completeness, V-measure, Adjusted Rand-Index and Silhouette Coefficient. These evaluation measures give value between 0 – 1. When these measures give value one cluster formed with good accuracy. Then the Predicted output will be correct. Cluster size can be increased. In this system cluster size upto 200 words.

Different set of contexts used for evaluation. For disambiguation purpose 50 different words are founded and make folder for each word. Set of contexts also created for prediction. It strictly contains 1678 features. Then it tested with data set. Evaluation measures gives value 1, system performed with 100 percent accuracy. That is system predict, test context related to which cluster. Other circumstances system performance not up to 100 percent. The folder content mixed up, that is a folder contains different type of disambiguated contexts. In this situation also system gives required output. In this situation evaluation measures gives low value. For getting good in evaluation measures separately gives the contexts corresponding folders.

V. CONCLUSION

Resolving ambiguity of words for getting the correct sense of the word in the context is word sense disambiguation (WSD) . WSD have a different context model for each individual word. Discriminate the word meanings based on information found in unannotated corpora is unsupervised WSD. One of the unsupervised WSD approaches is Context Clustering. Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects. The context Clustering method is based on clustering techniques in which first context vectors are created and then they will be grouped into clusters to identify the meaning of the word. Here, context vectors created from different contexts. Then grouped this context vectors into different clusters. Then give a WSD context as input , it mapped into the related cluster. Predicted the context belongs to which cluster. It can be applied so many natural language applications such as news classification, email sorting etc.

Future work:

To reduce the size of input without effecting the performance of the system. Label the clusters and predict cluster labels without effecting the performance. Reduce the number of feature taken and form more effective clusters.

ACKNOWLEDGMENT

Extremely grateful to Mr. Vishal Yadav, System Design Lead (AI), CogniCor Technologies,Cochin for the constant support and the cooperation throughout the work. Thankful to Mr. Gopalakrishnan G, NLP Engineer, Mr. Dibin Joseph Research Engineer and Jyothis K S, Indian Operations Lead CogniCor Technologies, Cochin for his encouragement and sincere guidance .

REFERENCES

[1] Alok Ranjan Pal1 and Diganta Saha, ”Word Sense Disambiguation: a Survey” International Journal of Control Theory and Computer Modeling (IJCTCM) Vol.5, No.3, ,July 2015.

[2] Cheng Niu, Wei Li, Rohini K. Srihari, Huifeng Li, Laurie Crist, ”Context Clustering for Word Sense Disambiguation Based on Modeling Pairwise Context Similarities” SENSEVAL-3: Third international Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, July 2004.

[3] Amruta Purandare and Ted Pedersen, ”Sense Clusters Finding Clusters that RepresentWord Senses” National Science Foundation Faculty Early Career Development award , 2004.

[4] http://scikitlearn.org/stable/documentation.html

[5] Henry Anaya-Sanchez, Aurora Pons-Porrata, and Rafael Berlanga- Llavori, ”Using Sense Clustering for the Disambiguation of Words” SemEval:International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, 2007.

[6] Tamara Martín-Wanton and Rafael Berlanga-Llavori “A clustering-based Approach for Unsupervised Word Sense Disambiguation” Procesamiento del Lenguaje Natural, Revista 2012.