ISSN: 2321-7782 (Online)
Volume 3, Issue 10, October 2015
International Journal of Advance Research in
Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

An Integration of Supervised and Unsupervised Machine Learning Algorithms to Optimize Word Sense Disambiguation

Neetu Sharma[1]
Mewar University, Chittorgarh, India

Prof. S. Niranjan[2]
Mewar University, Chittorgarh, India

***Abstract:**Natural language processing (NLP) is the ability of a computer program or software to understand the human words or speech as it is pronounced. NLP is a main component of artificial intelligence (AI). Current approaches to NLP are mainly based on machine learning techniques which is further a type of artificial intelligence (AI) that examines and exploit patterns in data to improve the program's own understanding. A number of different classes of machine learning algorithms have been applied to NLP tasks.Word sense disambiguation is the process to find best sense of ambiguous word from the existing senses to remove the ambiguity. Most commonly supervised machine learning algorithms were used to solve this problem and improve the performance. Some attempts were made to use unsupervised machine learning algorithms also like K-means clustering algorithm.*

In this paper supervised learning algorithm Naive Bayesian is combined with the unsupervised learning algorithm K-means Clustering and the performance is enhanced in getting best sense of ambiguous word. C# is used to create interface for putting input in the form of sentence containing ambiguous word and displaying the output as a best sense for that ambiguous word. SQL 2008 is used as a database to store the sentences entered and their corresponding meanings. The main contribution of the work is a completely new framework for word-sense disambiguation with a combination of supervised and unsupervised learning technique utilizing WordNet. WordNet is used as a database for extracting senses of ambiguous word. Performance is evaluated on the basis of scores of precision, recall and F-score that how well this optimized algorithm works to improve the accuracy.

***Keywords:** NLP, Naive Bayesian Algorithm, K-Means Clustering, WordNet, SQL 2008*

I. Introduction

Natural language processing[23] is an interesting and motivating domain in which to develop, maintain and evaluate representation and reasoning theories. All the problems of Artificial Intelligence comes in this domain like “solving the problem natural language problem" is as difficult as solving "the AI problem" because any domain can be expressed in natural language. The field of computational linguistics has a wealth of techniques and knowledge.

If we need a computer system to communicate with the user in their own terms; we would rather don’t need the user to learn a new language. This is especially important for normal users and those users, such as managers and children, who don’t have time to learn new interaction skills.

When there is a huge collection of information recorded in natural language that could be easily accessible via computers. As the information is constantly transformed in the form of books, newspapers, business reports and government reports, and scientific papers, many of which are available online. A system which requires a great deal of information must be able to process natural language to retrieve maximum information available on computers. Here are three important ways of understanding theory of natural language:

Syntax

The syntax is defined as the method of expressing a computer language. Always some rules of grammar are used to express it. Natural language is difficult to understand as compared to formal languages used for the logics and computer programs.

Semantics

The semantics are defined as the meaning of the sentences or words of the language used in the grammar. To create a system for understanding natural language for an application some semantic rules must be considered, and it is tried to use the simplest way of representation that can be possible. For example, in the development that follows, there is a fixed mapping between words and concepts in the knowledge base, which is inappropriate for many domains but simplifies development.

Pragmatics

The pragmatic part of natural language tells us about the words or sentences related to the real world. To understand natural language, a user must not only the sentence but its context words also. Also it includes the state of the world, the goals of the speaker and the listener, special conventions, and many more for consideration to understand.

1.1 Word Sense Disambiguation

Word sense disambiguation (WSD)[15] is the task to determine which meaning of a polysemous word is correct in a given context. The words which can have a no. of senses are called Polysemous Words.

For example, consider the word “bass”, it has two distinct meanings:

1. a kind of fish

2. low frequency tones

“I'm not sure if it's meant to be a guitar cab or a bass cab, but it suits me”

“Whitefish, bass, trout and pickerel are an important food supply obtained from the waters of the lake”.

To a human it is obvious the first sentence is using the word “bass” in sense 2 above, and in the second sentence it is being used in sense 1.

Although this seems obvious to a human, to develop algorithms to solve this ambiguity problem is a difficult task. In computational linguistics, word-sense disambiguation (WSD) is an open problem of natural language processing, which governs the process of identifying which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple meanings. The solution to this problem put impact on other computer-related writing matter, such as translation, improving relevance of search engines, parsing, coherence, inference etc.

WSD task has two ways: "lexical sample" and "all words" task. The “lexical sample” means disambiguating the words from a small sample which were already known, while in “all words” words disambiguated from the text while it is executing. The latter is called a more realistic form of evaluation, but the problem is that the corpus is much costly to produce because human translators have to read the definitions for each word in the sequence every time they need to make a decision, rather than once for a block of instances for the same target word.

There are four conventional approaches to WSD:

Dictionary- and knowledge-based methods: These include primarily the dictionaries, thesauri, and lexical knowledge bases, without using any corpus database.

Semi-supervised or minimally supervised methods: These make use of a secondary source of knowledge such as a small annotated corpus as seed data in a bootstrapping process, or a word-aligned bilingual corpus.

Supervised methods: These make use of sense-annotated corpora for training the machine and learning the machine.

Unsupervised methods: These uses completely explicit information and work directly from raw corpora that is unannotated. These methods are also known under the name of word sense discrimination.

1.2 Applications of WSD

Machine translation

Machine translation is the first most important and preferred application for WSD because WSD has been considered in almost every application of language technology, including information retrieval, knowledge mining/acquisition lexicography, and semantic interpretation, and is becoming increasingly important in new research areas such as bioinformatics and the Semantic Web also.

Information retrieval

In some query systems, ambiguity has to be resolved. For example, given the query _bank_ should the system return financial institution, river side, or collection of money? Today's IR systems such as Web search engines, like Machine Translation, do not use a WSD module; they give best result only when the user will type enough context words in the query to retrieve relevant documents. In mutual disambiguation method, all the ambiguous words are disambiguated only when the occurrence of senses of words are at same place.

The utility of WSD

WSD as a single module has not yet been used to make an effective difference among the applications. There are a few recent results that show small positive effects in, for example, machine translation, but WSD does not perform well as is the case in well-known experiments in information retrieval(IR). There are many reasons for the poor performance. First, the domain of word sense is very small and limited which an application requires (e.g., no one wants to see the tones of frequency sense of bass in a kind of fish sense), and so lexicons are being constructed accordingly. Second, is the accuracy, WSD is not much accurate enough to perform better and more over the sense inventory used is unlikely to match the specific sense distinctions required by the application. Third, seeing WSD as an individual component or module may be not true, as it is more tightly integrated as an internal process. It performs better only in integrated form.

Information extraction and knowledge acquisition

In information retrieval and text mining, WSD is required for the accurate analysis of data in many applications. For example, an intelligence collection system might be needed to attach references to, say illegal driving, rather than proper driving. Bioinformatics research requires the relationships between genetic and genetic products to be retrieved from the vast scientific literature; however, genes and their proteins often have the same name. More generally, the Semantic Web requires automatic annotation of documents according to reference ontology. WSD is to be applied in these areas is at the beginning stage.

1.3 Machine learning

Learning[28] can be defined as “any change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population”. Depending on the amount and type of knowledge available to the system before the learning phase (system’s a priori knowledge) it can be categorized in several situations:

» The first and simplest form of learning is the situation when the full knowledge is available that is required for a particular type of task.

» Second type of learning is to store the data in the similar format and it is called rote learning. For example filling a database.

- Third type is the process of knowledge acquisition in an expert system which is a kind of learning task where some pre-defined structures (rules, frames etc.) are filled with data specified directly or indirectly by an expert. In this case only the structure of the knowledge is known.
- Fourth type is the system in which a set of examples (training data) is given and it is required to generate a description of this set in terms of a particular language. This is an advance knowledge of the system which is the syntax of any known language on synctactic basis. Possibly some characteristics of the domain from which the examples are drawn are taken(domain knowledge or semantic bias). This is a typical task for Inductive learning and is usually called Concept learning or Learning from examples.
- Another current trend's type is learning systems is Neural networks which does not given a knowledge at prior and can react properly to the text. Neural networks actually use a kind of a pre-defined structure of the knowledge to be represented (a network of neuron-like elements), which however is very general and thus suitable for various kinds of knowledge. As in human learning the process of machine learning is affected by the presence (or absence) of a teacher. In the supervised learning systems the teacher explicitly specifies the desired output (e.g. the class or the concept) when an example is presented to the system(i.e. the system uses pre-classified data). In the reinforcement learning systems the exact output is unknown, rather an estimate of its quality (positive or negative) is used to guide the learning process. Conceptual clustering (category formation) and Database discovery are two instances of Unsupervised learning. The aim of such systems is to analyze data and classify them in categories or find some interesting regularities in the data without using pre-classified training examples.

1.3.1 Types of machine learning algorithms

After the understanding of the type of machine learning applications working with, now the type of data to collect and the types of machine learning algorithms must be discussed. It is useful to discuss the main algorithms to get a general idea of what methods are available. There are a number of algorithms available. There are classes of method and there are extensions also further to these methods and it usually becomes difficult to find out what makes a specific algorithm, this is the main problem.

There are so many ways to classify machine learning algorithms,[20] first is a grouping of algorithms by the learning style, second is a grouping of algorithms by similarity in form or function. Both approaches are useful. An algorithm can be modelled in a number of different ways based on its interaction with the environment or input data or experience.

An algorithm can have a few types of learning methods as follows:

Supervised Learning: This method has two parts one is training data and another is testing data. Generally input data is specified as training data along with a known label or result such as spam/not-spam. The model of this algorithm is prepared through a training process along with the predictions and to correct those predictions when wrong. The training process repeats till the model achieves a desired level of accuracy on the training data. Example problems are classification and regression. Example algorithms are Decision Tree, Support vector machine, Naive Bayesian Algorithm.

Unsupervised Learning: This algorithm has only part that is testing data that is called input data which is not labelled and does not have any known result. Its model is created by using the structures present in the input data. Example problems are association rule learning and clustering. Example algorithms are the K-means Clustering algorithm and apriori algorithm

Semi-Supervised Learning: In this algorithm input data is a group of labelled and unlabelled data. In this the model can use the structures to organize the data as well as can make assumptions also. Example problems are classification and regression. Example algorithms are combinations to other algorithms which can be modified that make assumptions about how to model the unlabelled data.

Reinforcement Learning: In this algorithm input data is inputted as a stimulus to a model from an external source or environment to which the model must respond and react. Feedback is not provided from a teaching process as in supervised learning, but as punishments and rewards in the environment. Example problems are systems and robot control. Example algorithms are Q-learning and Temporal difference learning.

Transduction: It is almost similar to supervised learning but it does not explicitly construct a function yet it tries to assume new outputs based on training inputs, outputs and new inputs.

II. Motivation And Research Objectives

WSD is one of the most important open problems in the Natural Language Processing (NLP) field. There is the wide range of approaches which were investigated and the large effort devoted to handle this problem, it is a fact that till date no large scale broad coverage and highly accurate word sense disambiguation system has been built. The main reasons why we have chosen WSD are:

- There is infrastructure to help you get started.
- It is accessible to anyone with an interest in NLP.
- It's an interesting problem.
- It persuades you to work on word sense disambiguation
- Lots of good work already done, still more to do.
- Persuade you to use word sense disambiguation in your text applications.

Machine learning is a branch of artificial intelligence which studies mechanisms to mimic the ability of humans to learn. Machine learning strives to get the computer to learn tasks such as discriminating between objects, segregating similar objects from dissimilar ones and learning from experience. There are so many Machine Learning (ML) methods or algorithms have been used to produce more successful Word Sense Disambiguation (WSD) systems. There are still some differences among the performance measurements of different algorithms which are not evaluated; hence it can be used for further detailed investigation for the task. These tasks are formally known as supervised, unsupervised and reinforcement learning in the machine learning parlance. In supervised learning, the system is presented with a set of data which is labeled into various categories and involves learning a function which maps the data to the categories.

III. Literature Review

Azzini, C. da Costa Pereira, M. Dragoni, and A. G. B. Tettamanz[28] proposed a supervised approach to word sense disambiguation based on neural networks combined with some more algorithms. They have taken large datasets for every polysemous word senses and used some optimization method for neural network that has correctly disambiguates the sense of the given word by taken the context words in which it occurs into consideration. The feasibility of the approach has been shown through experiments carried out on a particular set of input polysemous words.

Rion Snow Sushant Prakash, Daniel Jurafsky, Andrew Y. Ng[29] formulated a new method of merging of senses as a supervised learning problem, by using manually tagged sense clustering as training data. The data for training a disambiguating classifier has been derived from WordNet database, corpus-based proof data, and evidence from other lexical resources. The similarity measure performs much better than previously proposed automatic methods for sense clustering on the task of predicting human sense merging judgments, which yields an absolute F-score improvement of 4.1% on nouns, 13.6% on verbs, and 4.0% on adjectives. Finally, a model is devised for clustering sense taxonomies using the outputs of the classifier, and it is e automatically clustered for senses taking data from WordNets.

Dinakar Jayarajan [31] presented a new representation for documents based on lexical chains. Their work includes both the problems and achieves a better reduction in the dimensionality and results in the semantics as output present in the input data. They devised an optimized algorithm to compute lexical chains and generate feature vectors using these chains.

Yee Seng Chan and Hwee Tou Ng, David Chiang [30]presented an experimental study to state that word sense disambiguation (WSD) systems can help to improve the performance of statistical machine translation (MT) systems. They successfully integrated a state-of-the-art WSD system into a state-of-the-art hierarchical phrase-based MT system. They presented first time that integrating a WSD system improves the performance of a state-of-the-art statistical MT system on an actual translation task.

S.K.Jayanthi and S. Prema [32] performed a number of investigations into the relationship between information retrieval (IR) and lexical ambiguity in web mining. The work is much exploratory. The results of these experiments lead to the conclusions that query size plays an important role in the relationship between ambiguity and IR in web content mining. Word Sense Disambiguation (WSD) is tested and analyzed for some of the existing Information Retrieval engines like Google, MSN, yahoo, Altavista search using Brills tagger, and the derived results for the IR systems recommends how to accommodate the sense information in the selected document collection.

IV. NAIVE BAYESIAN ALGORITHM

A Naive Bayesian classifier is a simple classifier works on probabilistic theory from Bayesian statistics including independent assumptions. This classifier can also be called as "independent feature model"

In general, a Naive Bayes classifier assumes that the existence or non existence of a particular feature of a class is independent of presence or absence of any other feature in the class. For example, a fruit can be considered to be an orange if its color is orange, it is round in shape, and about 4" in diameter. If these features depend on each other or upon the existence of the other features, a Naive Bayes classifier considers all of these features to independently declare with the probability that this fruit is an orange.

As a supervised learning algorithm, Naive Bayes can be trained very efficiently using probabilistic model or theory. Practically, in almost many applications, estimation for parameters in Naive Bayes models uses the method of maximum likelihood; in other words, we can say that working with the Naive Bayes model is more simpler by not believing in any Bayesian probability methods.

As the design and structure of Naïve Bayesian classifier is very simple still Naive Bayes classifiers worked very well in all complex real-world situations. The main benefits draws from the Naive Bayes classifier is that it needs only a small amount of training data to estimate the performance necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. This is a simple probabilistic classifier based on the Bayes theorem.

$$P(Y/Z) = \frac{P(Z/Y)P(Y)}{P(Z)} = \frac{P(Z/Y)P(Y)}{\big(P(Z/Y)P(Y) + P(Z/X)P(X)\big)}$$

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

In the Bayesian interpretation or theory, the probability measures a degree of belief. Bayes' theorem links the degree of belief in a proposition before and after accounting for evidence. For example, suppose somebody proposes that a biased coin is twice as likely to land heads as tails. Degree of belief in this might initially be 50%. The coin is then flipped a number of times to collect evidence. Belief may rise to 70% if the evidence supports the proposition. For proposition A and evidence B, P(A), the prior, is the initial degree of belief in A.

P(A | B), the posterior, is the degree of belief having accounted for B.

P(B | A) / P(B) represents the support B provides for A.

4.1 Algorithm:

1) Preprocessing

a. Segmentation of words in inputted

sentence

b.Remove stop words, punctuations etc.from

inputted sentence

2) Multi sense lookup

Find out possible meanings or senses of the ambiguous word from the database

3) Calculation of Probability

for all senses si of ambiguous word A do

or

all words fi in the vocabulary do

P(fi|si) = C(fi,si)/C(si)

end

end

for all senses si of A do

P(si) = C(si) / N

end

4) Disambiguation process

for all senses si of A do

score(si) = log P(si)

for all words fi in the context window c do

score(si) = score(si) + log P(fi|si)

end

end

Choose s' = arg max score(si)

4.2 Benefits:

Naive Bayes based on the independent assumptions as:

» Training requires each attribute in each class independently makes it faster and easier.

» Testing is also simpler by calculating data from the tables directly and using conditional probabilities with normal distribution.

» Naive Bayes looks as a common generative model

» It has performance similar to most of the state-of-the-art classifiers even in presence of violating independent assumption

» It gives best performance for the spam mail filtering application.

» A good candidate of a base learner in ensemble learning

» Naive Bayes can do some other functions also along with simple classification.

V. K-MEANS CLUSTERING ALGORITHM

K-means clustering algorithm is the simplest unsupervised learning algorithms that solve the most of the clustering problem. It has a simple and easier way to make clusters in a given data set using classification method of clusters (assume k clusters) fixed in advance. The main process starts with defining k centroids, selecting one for each cluster. These centroids should be placed in a different location because of different location causes different result. So, the better option is to place the centroids very far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is left then the first step is completed and an first stage grouping is done. In next step again k new centroids are calculated which act as bar centers of the clusters resulting from the previous step. After getting k new centroids, a new grouping has to be performed between the same data set points and their nearest new centroids. Doing this a loop has been generated and it has been noticed that the k centroids change their location one by one until no more changes are required or in other words centroids do not move any more. This non hierarchical method initially takes the number of components of the population equal to the final required number of clusters. In this step the clusters are chosen based on finally required number of clusters such that the points are mutually farthest apart. In next step, it examines each component in the population and assigns its value to one of the clusters depending on the least distance from the centroid. The centroid's position is recalculated every time a component is added to the cluster and this continues until all the components are grouped into the final required number of clusters. Finally, the aim of this algorithm is to minimize an objective function called a squared error function.

5.1 Algorithm

1. Selection of Initial Value of Centroids

From the given set of examples initial centroid values are selected at random.

2. Calculate Objects - Centroid Distance

The distance between each cluster centroid to each object is calculated and recorded. The measure used for this calculation is Euclidean distance measure. After this step a distance matrix is generated this represents the distance between each object and each cluster centroid

3. Assigning Object to cluster

Objects are assigned to a specific cluster based on a minimum distance measure. That is, for any given object ‘A’, if the distance between this object and the centroid of cluster n is minimum, then object ‘A’ will be assigned to cluster n.

4. Iteration 1 : Determine Centroids

Once the members of each group are determined, the new centroids can be computed based on these new memberships. For example, if a group (cluster) has 3 members, the centroid of this group will be given by the average of the co-ordinates of the 3 group members.

5. Iteration 1 : Object - Centroid Distances

Similar to step 2, the next step involves calculating the distance of each object to the new centroids, once again generating a distance matrix.

6. Iteration 1 : Object Clustering

Similar to step 3, objects are assigned to groups based on the minimum distance criteria.

7. Iteration 2 : Determine centroids

The whole process is carried out iteratively until the centroid values become constant, i.e. do no change iteratively.

5.2 Benefits

1) Simple design, Fast, robust and easier to implement and understand.

2) More efficient as compared to other clustering algorithms.

3) It performs best when data set are distinct or far separated with each other.

VI. WORDNET

WordNet is a lexical set database of words having more than one meaning or we can call them synonymous words. It has a large vocabulary of nouns, verbs, adjectives and adverbs. If the word belongs to any of the category then it will display the corresponding senses from the database. It is mainly supported by the National Science Foundation under Grant Number 0855157. NSF is fully responsible for any changes, views etc..

The vocabulary of any language is defined as a set X of pairs (a,s), where a is a string over a finite alphabet, and a sense s is an element from a given set of senses. Forms can be utterances composed of a string of phonemes or inscriptions composed of a string of characters. Each form with a sense in a language is called a word in that language. An alphabetical ordered list of words is called a dictionary. A word having more than one sense is called polysemous word; two words that have at least one sense in common are said to be synonymous. Set C is set of contextual words in which the word can be used. The set C of language partitions it into syntactic categories. Words that occur in the subset N are nouns; words that occur in the subset V are verbs, and so on. There is a hierarchical structure for each category of syntactic contexts. The set of contexts in which a particular string a can be used to express a particular sense s.

In WordNet, a form is represented by a string of ASCII characters, and a sense is represented by the set of (one or more) synonyms that have that sense. WordNet contains more than 118,000 different word forms and more than 90,000 different word senses, or more than 166,000 (a,s) pairs. Approximately 17% of the words in WordNet are polysemous; approximately 40% have one or more synonyms. WordNet respects the syntactic categories noun, verb, adjective, and adverb—the so-called open-class words. For example, word forms like “bank,’’ “right,’’ or “interest’’ are interpreted as nouns in some linguistic contexts, as verbs in other contexts, and as adjectives or adverbs in other contexts; each is entered separately into WordNet. It is assumed that the closed-class categories of English some 300 prepositions, pronouns, and determiners play an important role in any parsing system; they are given no semantic explication in WordNet. Inflectional morphology for each syntactic category is accommodated by the interface to the WordNet database. Synonymy is WordNet’s basic relation, because WordNet uses sets of synonyms (synsets) to represent word senses. Synonymy (syn same, onyma name) is a symmetric relation between word forms.

- Antonymy (opposing-name) is also a symmetric semantic relation between word forms, especially important in organizing the meanings of adjectives and adverbs.
- Hyponymy (sub-name) and its inverse, hypernymy (super-name), are transitive relations between synsets. Because there is usually only one hypernym, this semantic relation organizes the meanings of nouns into a hierarchical structure.
- Meronymy (part-name) and its inverse, holonymy (whole-name), are complex semantic relations. WordNet distinguishes component parts, substantive parts, and member parts.
- Troponymy (manner-name) is for verbs what hyponymy is for nouns, although the resulting hierarchies are much shallower.

VII. Performance Measures

Once a predictive model is implemented, it is always required to know that whether it is working good or not. If it is working well then we want to know the level of goodness or we can say how effectively it is fulfilling the goal of the task. To know all these things we must be aware about the metrics used to measure the performance. Commonly the performance measures used to evaluate the performance at work are:

Accuracy:

Accuracy is the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications.

Precision:

Precision is a measure of the accuracy provided that a specific class has been predicted. It is defined by:

Precision = tp/(tp + fp)

where tp and fp are the numbers of true positive and false positive predictions for the considered class. In the confusion matrix above, the precision for the class A would be calculated as:

Precision(A) = tp(A)/(tp(A)+eBA+eCA) = 25/(25+3+1) ≈ 0.86

The number is reported by RDS as a value between 0 and 1.

Recall:

Recall is a measure of the ability of a prediction model to select instances of a certain class from a data set. It is commonly also called sensitivity, and corresponds to the true positive rate. It is defined by the formula:

Recall = Sensitivity = tp/(tp+fn)

where tp and fn are the numbers of true positive and false negative predictions for the considered class. tp + fn is the total number of test examples of the considered class. For class A in the matrix above, the recall would be:

RecallA = SensitivityA = tpA/(tpA+eAB+eAC) = 25/(25+5+2) ≈ 0.78

F-Score:

In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and there call r of the test to compute the score: p is the number of correct results divided by the number of all returned results and r is the number of correct results divided by the number of results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0.

The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

VIII. Results And Discussions

Results found after performing implementation are shown below. It includes implementation of optimized algorithm and sees the results step by step. The database used is a large lexical dataset called WordNet which has a large vocabulary of data set including nouns, verbs, adjectives and adverbs.. The interface is created in C# language to enter the sentence with ambiguous word and displaying the result as best sense out of all senses retrieved from the WordNet database. SQL2008 is used to store the results after getting senses from the WordNet. Since individual algorithms produce diverse results in terms of precision that complement each other well in terms of coverage. A combined approach outperforms score of best individual classifier. The two main algorithms named as Naive Bayesian Algorithm(Supervised) and K-Means Clustering algorithm(Unsupervised) are combined to form an optimized process to improve the performance of word sense disambiguation. At last performance measures are shown in tabular form to compare the results with previous techniques which are used to evaluate the performance. F-Score measure value increases as on applying optimized algorithm on input data.

We implement our work using followings steps:

We create a user interface for entering the sentence at run time in C# language.

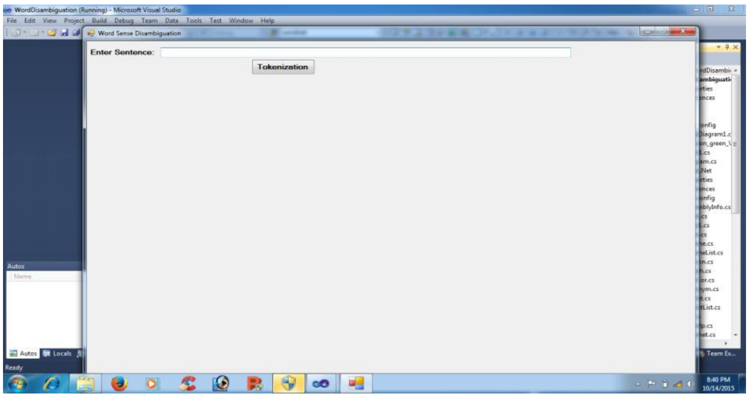


Fig. 7.1 User Interface

We create a module for tokenization of the entered sentence for selection of ambiguous word and its context words.

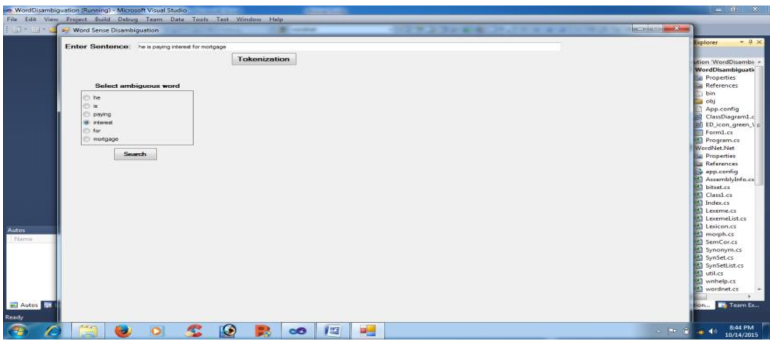


Fig. 7.2 Tokenization

We select an ambiguous word from the created tokens

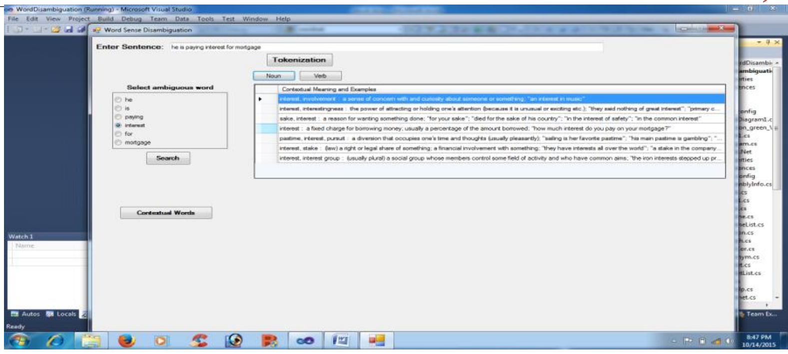


Fig. 7.3 Selecting an ambiguous word

We create a class to access the WordNet API database for displaying the all possible meaning of selected word as a noun.

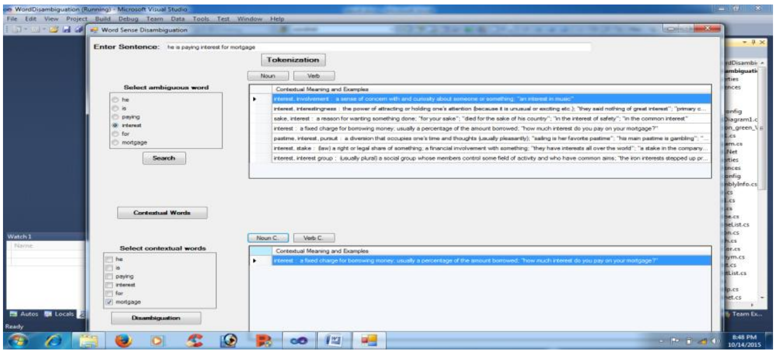


Fig. 7.4 WordNet Access for Noun

We create a class to access the WordNet API database for displaying the all possible meaning of selected word as a verb.

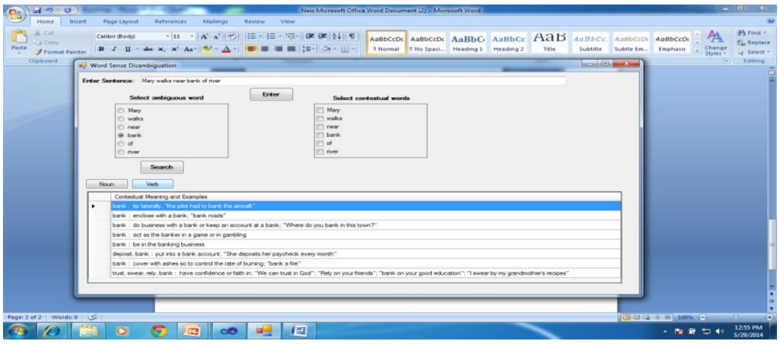


Fig. 7.5 WordNet access for verb

We create a module for K-Means clustering algorithm to create clusters of different meanings of ambiguous word based on context word and store it in the WordDisambiguation table

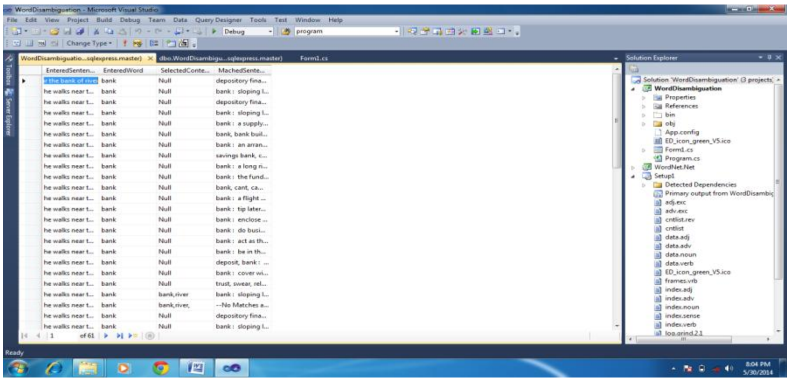


Fig. 7.7 Word Disambiguation table

We create a module NBClassifier for Naïve Bayes classifier, which takes clusters as input.

We create a submodule also in NBClassifier module for learning the Naïve Bayes classifier.

We create a submodule in NBClassifier module for disambiguation of selected ambiguous word from its context.

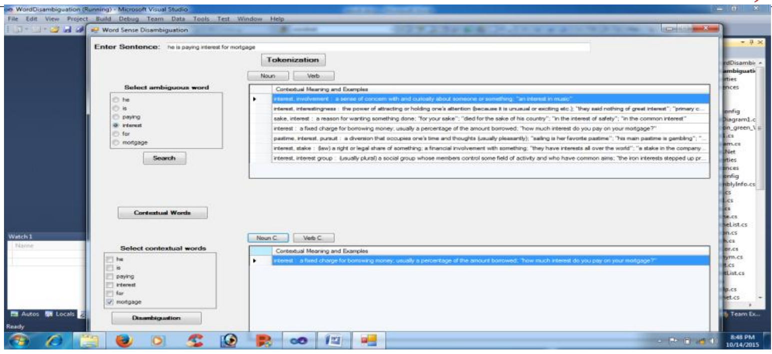


Fig.7.8 Final Output

We take different words to perform above tasks and noted the output.

We Calculate the precision-recall, F-Score and Accuracy from above results.

Words	Part of speech tagging	No of sentences	Correctly Identified Sense	Incorrectly Identified Sense	Not identified
interest	Noun/Verb	10	8	2	0
Bass	Noun /Adj	8	3	3	4
bank	Noun/Verb	7	4	0	3
Cut	Verb/Noun/Adj	7	4	1	2
Play	Noun/Verb	14	9	2	0
step	Noun/Adj/ Verb/Advrb	12	10	2	1

Table 4 Measurement of parameters for 6 words

This above table shows the measurement of the system. In this table we have five cases these are given below:

Part of speech Incorrectly identified Not identified

No of sentences Correctly identified

By checking of all these cases for taking different examples like: bank, step, master, bass, etc. all these examples give the measurement of the system. By the help of above table we have also measure the performance of system.

For calculation of our system: A = Correct identified sentence B = incorrect identified sentence

C = Not identified sentence Recall = (A/ (A+B))*100% Precision = (A/ (A+C))*100%

IX. Conclusions And Future Scope

The main aim of the research work is to improve the performance of word sense disambiguation for English language words like nouns, verbs, adverbs and adjectives. We use a combinatorial approach to fulfill this aim. In previous research only a single machine learning classifier was used to disambiguate the words. The classifier selected the words by means of a previous analysis of training data in order to identify which ones seem to be highly accurately disambiguated. By combining two algorithms Naïve Bayesian and K-means clustering performance is enhanced for disambiguation of words in English. Precision, recall and F-Score values are also improved which shows that now the disambiguation process gives more accurate results then previous methods. The following are the most interesting issues that have something to do in future also, and which adds further investigation based on the findings of this thesis:

» The same combinatorial approach can be used for some another language also.

- » The optimized approach can be used for some other applications of NLP like machine translation, information retrieval, parsing etc.
- » Some other newer algorithm for classification can be used to enhance the performance for WSD.
- » Dataset can be changed; it can be performed on some other data source also.
- » In future disambiguation of sentences can be done by considering all parts of the sentence as context words.
- » There is a wide scope for future in NLP generating more application areas and their problem solving.

References

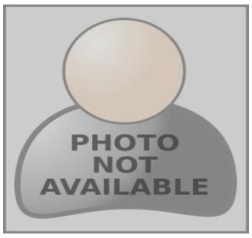
1. Kilgarriff, A., Evans, M. (eds): Special issue on SENSEVAL. Computer and the Humanities, 34(1-2) (2000)
2. Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Five Papers on WordNet. Special Issue of International Journal of Lexicography, 3(4), (1990)
3. Miller, G. A., Leacock, C., Tengi, R., Bunker, R.T.: A Semantic Concordance. Proceedings of the ARPA Workshop on Human Language Technology (1993)
4. Hirst, G.: Semantic Interpretation and the Resolution of Ambiguity. Cambridge University Press, England (1987)
5. McRoy, S.: Using Multiple Knowledge Sources for Word Sense Discrimination. Computational Linguistics, 18(1) (1992)
6. Ide N., Veronis J.: Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. Computational Linguistics, 24(1998)
7. Bruce, R., Wilks, Y., Guthrie, L., Slator, B., Dunning, T.: NounSense - A Disambiguated Noun Taxonomy with a Sense of Humour. Research Report MCCS-92-246. Computing Research Laboratory, New Mexico State University (1992)
8. Rigau, G., Atserias, J., Agirre, E.: Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. Proceedings of ACL-EACL, Madrid, Spain. (1997)
9. Resnik, P.: Selection and Information: A Class-Based Approach to Lexical Relationships. Ph.D. University of Pennsylvania (1993)
10. Agirre, E.: Formalization of concept-relatedness using ontologies: Conceptual Density. Ph.D. thesis. University of the Basque Country (1999)
11. Yarowsky, D.: One Sense per Collocation. Proc. of the 5th DARPA Speech and Natural Language Workshop (1993)
12. Ng, H. T., Lee, H. B.: Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-based Approach. Proceedings of the ACL (1996)
13. Leacock, C., Chodorow, M., Miller, G. A.: Using Corpus Statistics and WordNet Relations for Sense Identification. Computational Linguistics, 24(1) (1998)
14. Agirre, E., Martinez, D.: Exploring automatic word sense disambiguation with decision lists and the Web. Proceedings of the COLING Workshop on Semantic Annotation and Intelligent Content. Saarbrücken, Germany (2000)
15. Samit Kumar, Neetu Sharma, Dr. S. Niranjan," Word Sense Disambiguation Using Association Rules: A Survey", International Journal of Computer Technology and Electronics Engineering Volume 2, Issue 1, ISSN 2249-6343 (2011)
16. Mihalcea, R., Moldovan, D.: Word Sense Disambiguation based on Semantic Density. Proceedings of COLING-ACL Workshop on Usage of WordNet in Natural Language Processing Systems. Montreal, Canada (1998)
17. Yarowsky, D.: Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. Proceedings of COLING. Nantes, France (1992)
18. Yarowsky, D. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. Proceedings of the ACL. Cambridge, USA (1995)
19. Agirre, E., Martinez, D.: Learning class-to-class selectional preferences. Proceedings of the ACL CONLL Workshop. Toulouse, France (2001)
20. Martinez, D., Agirre, E.: Word Sense Disambiguation using syntactic cues. Internal Report. University of the Basque Country (forthcoming)
21. Yarowsky, D.: Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. Proceedings of the ACL (1994)
22. Neetu Sharma, Samit Kumar, Dr. S. Niranjan," Using Machine Learning Algorithms for Word Sense Disambiguation: A Brief Survey", International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 1, ISSN 2249-6343 (2011)
23. Agirre, E., Rigau, G.: Word Sense Disambiguation using Conceptual Density.. Copenhagen, Denmark (1996)
24. Neetu Sharma, S. Niranjan," An Optimized Combinatorial Approach of Learning Algorithm for Word Sense Disambiguation", International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064(2014)
25. Neetu Sharma, Dr. S. Niranjan," Optimization of word sense disambiguation using clustering in weka", Int.J.Computer Technology & Applications,Vol 3 (4), 1598-1604, ISSN:2229-6093, IJCTA,July-August(2012)
26. Neetu Sharma, Dr. S. Niranjan," Performance Enhancement Using Combinatorial Approach of Classification and Clustering In Machine Learning", International Journal of Application or Innovation in Engineering & Management (IJAIEM), ISSN 2319 – 4847,Volume2,Issue4,(2013)
27. Azzini A.,Costa D. Pereira M., Dragoni, and A. G. B. Tettamanzi, "Evolving Neural Networks for Word Sense Disambiguation", Eighth International Conference on Hybrid Intelligent Systems,(2008).
28. Snow Rion Prakash Sushant ,Jurafsky Daniel , Andrew Y. Ng ,"Learning to Merge Word Senses", Computer Science Department Stanford University,(2007).

29. Yee Seng Chan and Hwee Tou Ng, David Chiang,"Word Sense Disambiguation Improves Statistical Machine Translation", Department of Computer Science National University of Singapore, (2007).

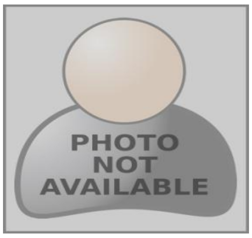
30. Jayarajan Dinakar ,"Using Semantics in Document Representation: A Lexical Chain Approach ", Department of Computer Science and Engineering Indian Institute of technology Madras, June (2009).

31. Jayanthi S.K., Prema S,"Word Sense Disambiguation in Web Content Mining Using Brill's Tagger Technique", International Journal of Computer and Electrical Engineering, Vol. 3, June (2011).

AUTHOR(S) PROFILE



Neetu Sharma, has completed BE(CSE) in 1996. and M. Tech (CSE) completed in 2006 from Panjab University. Presently she is doing Ph. D. in Computer Science and Engineering. Her total teaching experience is 16 years. She is presented 2 papers in National and seven papers in International conferences sponsored by IEEE**.** She has published several papers in International Journals.



Prof.S. Niranjan, did his M.Tech (Computer Engg.) from IIT Kharagpur in 1987. Completed Ph.D.(CSE) in 2004 and Ph.D.(I&CT) in 2007, Have total 26 years of teaching experience. Presently working as Professor in Mewar University, Chittorgarh, (Rajasthan). Having 16 publications in journals and 30 other papers published in conferences.