Word Sense Disambiguation by using Naïve Bayesian Classification

Thida San, Tin Myat Htwe
University of Computer Studies, Yangon
thidasan.ucsy@gmail.com,tinmyathtwe@gmail.com

Abstract

Natural Language Processing has been developed to allow human-machine communication to take place in a natural-language. Word Sense Disambiguation (WSD) is regarded as one of the most interesting and longest-standing problems in NLP. Several methodological issues come up with the context of WSD. These are supervised vs. unsupervised WSD approaches. Supervised WSD approaches have obtained better results than unsupervised WSD approaches. Naïve Bayesian WSD approach is one of the best supervised WSD approaches. This paper presents a corpus-based approach that uses Naïve Bayesian Classification to disambiguate ambiguous words with part-of-speech 'noun', which uses topical feature that represent co-occurring words in bag-of-word feature. This system also uses Senseval-3 corpus as a training data for Naïve Bayesian Classification, and access Word Net for retrieving meaning of the resulted senses. This system tokenizes and tags part-of-speech to each word of input sentence, collect target words, disambiguate target words and output correct sense and meaning for each target word.

1. Introduction

Word Sense Disambiguation (WSD) is one of the most critical and widely studied NLP tasks. Word Sense Disambiguation is the problem of assigning the appropriate meaning (sense) to a given in a text. Resolving the ambiguity of words is a central problem for language understanding applications and their associated tasks, including, for instance, machine translation, information retrieval and hypertext navigation, parsing, speech synthesis, spelling correction, reference resolution, automatic text summarization, etc.

Word Sense Disambiguation is often cast as a problem in supervised learning, where a disambiguator is induced from a corpus of manually-tagged text using methods from statistics or machine learning. These approaches typically represent the context in which sense-tagged instance of a word occurs with a set of linguistically motivated features. A learning algorithm induces a representative model from these features which employed as a classifier to perform disambiguation. Naïve Bayesian Classification has been used in many fields. Naïve Bayesian classifier greatly simplify learning by assuming that features are independent given class. Although independence is generally a poor assumption, in practice Naïve Bayesian Classification often competes well with more sophisticated classifiers. Naïve Bayesian Classification chooses the class (or sense) with the highest conditional probability for a target word.

2. Related Works

Word Sense Disambiguation is always a difficult and important task in natural language processing. Its task is to determine the most appropriate sense for an ambiguous word given a context. Approaches for this work include supervised learning, unsupervised learning, and combinations of them. Except for the expense involved in building labeled datasets, supervised based methods generally give results with higher precision. Many supervised learning algorithms have been applied, such as Bayesian learning, Exemplar-Based learning, Decision Trees, Decision Lists, and Neural Networks. Despite their simplicity, NB methods are still effective when applied to WSD. Mooney [4]compared six supervised algorithms including Naïve Bayesian Classification, Perceptron, Decision-Tree, k Nearest-Neighbor classifier, logic-based DNF (disjunctive normal form), and CNF (conjunctive normal form), and concluded that Naïve Bayesian Classification and Perceptron are the best methods for Word Sense Disambiguation. Pedersen [6] proposed a simple but effective approach using Ensembles of Naïve Bayesian classifiers to show that Word Sense Disambiguation accuracy can be improved by combining a number of simple classifiers into an ensemble. Leacock and Chodorow [3] used an Naïve Bayesian classifier, and indicated that by combining topic context and local context they could achieve higher accuracy. In comparing NB methods with Exemplar-Based methods, Escudero [1] utilized most of the features used in Ng and Lee [5], and showed that exemplar-based algorithm outperforms the Naïve Bayesian algorithm.

In many WSD studies, authors use Naïve Bayesian Classification as a baseline method for comparison, but many of them use Naïve Bayesian with only topic context while adding other information to their own methods.

3. Naïve Bayesian for Classification

A learning algorithm or induction algorithm is the forms of concept descriptions from example data. Concept descriptions are often referred to as the knowledge or model that the learning algorithm has induced form the data. Knowledge may be represented differently from one algorithm to another. For example, C4.5 represented knowledge as a decision tree; Naïve Bayes represented knowledge in the form of probabilistic summaries Naive Bayes Classification (NBC) is a machine learning method, particularly popular in medical applications. NBC assumes that the attributes are mutually independent. Although in practice this assumption is not quite true, experience shows that the NBC approach is effective and gives relatively good classification accuracy in comparison with other, more, elaborate learning methods.

3.1. Bayes' Theorem

Bayesian classifiers are statistical classifier. They can predict class membership probabilities, such as the probability that a given sample belongs to a particular class. Bayesian classification is based on Bayes' Theorem. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases. Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is known as "class conditional independence".

Let X be a data sample whose class label is unknown

Let H be a hypothesis that X belongs to class C

For classification problems, determine P(H/X): the probability that the hypothesis holds given the observed data sample X

P(H): prior probability of hypothesis H (i.e. the initial probability before we observe any data, reflects the background knowledge)

P(X): probability that sample data is observed

P(X|H) : probability of observing the sample X, given that the hypothesis holds

Given training data X, posteriori probability of a hypothesis H, P(H|X) follows the Bayes theorem:

$$P(H|X)=\frac{P(X|H)P(H)}{P(X)}$$

Informally, this can be written as

posterior = likelihood x prior / evidence

MAP (maximum posteriori) hypothesis:

$$h_{MAP} = \underset{h \in H}{\text{argmax}}\, P(h \mid D)\ \underset{h \in H}{\text{argmax}}\, P(D|h)P(h).$$

Practical difficulty: require initial knowledge of many probabilities, significant computational cost

3.2. Naive Bayesian Classifiers

A naive Bayes classifier is a term in Bayesian statistics dealing with a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model".

In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even though these features depend on the existence of the other features, a naïve Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods. Suppose that there are m classes, C1, C2, …, Cm. Given an unknown data sample, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naïve Bayesian classifier assigns an unknown sample X to the class C*i* if and only if

$$P(C \setminus X_i)\ P(C \setminus X_j) \quad \text{for } 1 \leq j \leq m \text{ and } j \neq i$$

Where

$$P(C_i \setminus X) = P(X \setminus C_i)P(C_i) / P(X)$$

$$P(X \setminus C_i) = \prod_{k=1}^{n} P(X_k \setminus C_i)$$

4. Naïve Bayesian Classification for Word Sense Disambiguation

4.1. Word Sense Disambiguation

A word sense is one of the meanings of a word. A word is called ambiguous if it can be interpreted in more than one way, i.e., if it has multiple senses. Disambiguation determines a specific sense of an ambiguous word. Word Sense Disambiguation

(WSD) is the process of selecting the appropriate meaning or sense for a given word in a document. WSD is one of the fundamental and important processes needed for many Natural Language Processing (NLP) applications, especially for language translation. Many WSD algorithms rely on contextual similarity to help choose the proper sense of a word in context. Several important methodological issues come up in the context of word sense disambiguation. These are:

- All words approach or unsupervised and
- Supervised or lexical sample approach

Many Word Sense Disambiguation approaches use the following as sources:

i. Dictionaries and thesauri
ii. Word Net
iii. Automatic, corpus-based; apply heuristics
iv. Variation or combination of above

4.1.1. Supervised Word Sense Disambiguation approach. In supervised disambiguation, a disambiguated corpus is available for training. There is a training set of exemplars where each occurrence of the ambiguous word w is annotated with a semantic label (usually its contextually appropriate sense sk). This setting makes supervised disambiguation an instance of statistical classification. The task is to build a classifier which correctly classifies new cases based on their context use ci. This notation, which we will use throughout the paper, is shown in Figure 1.

Symbol	Meaning
w	an ambiguous word
$s_1,\ldots,s_k,\ldots,s_K$	senses of the ambiguous word
$c1,\ldots,c_i,\ldots,c_I$	contexts of w in a corpus
$v1,\ldots,v_j,\ldots,v_J$	words used as contextual features for disambiguation

Figure 1.Notational conventions used in this paper

4.2. Bayesian classification

Bayesian classifier for word sense disambiguation is that it looks at the words around an ambiguous word in a large context window. Each content wore contributes potentially useful information about which sense of the ambiguous word is likely to be used with it. The supervised training of the classifier assumes that we have a corpus where each use of ambiguous words is labeled with its correct sense.

These context windows can be presented in two classes:

- Bag-of-word feature vectors – These are unordered set words with their exact position ignored.
- Collocation feature vectors – A collocation is a word or phrase in a position specific relationship to a target word.

A Bayes classifier applies the Bayes decision rule when choosing a class,

Decide s' if $P(s'|c) > P(s_k|c)$ for $s \neq s'$

We do not know the value of $P(s_k|c)$, but we can compute it using Bayes' rule,

$$P(s_k|c) = \frac{P(c|s_k)\,P(s_k)}{P(c)}$$

$P(s_k)$ is the prior probability of sense s_k, the probability that we have an instance of s_k if we do not know anything about the context. $P(s_k)$ is updated with the factor $P(c|s_k) / P(c)$ which incorporates the evidence which we have about the context, and results in the posterior probability $P(s_k|c)$.

If all we want to do is choose the correct class, we can simplify the classification task by eliminating P (c) (which is a constant for all senses and hence does not influence what the maximum is). We can also use logs of probabilities to make the computation simple. Then, we want to assign *w* to the sense s' where:

$$\begin{aligned} s' &= \arg\max_{s_k} P(s_k|c) \\ &= \arg\max_{s_k} \frac{P(c|s_k)}{P(c)} P(s_k) \\ &= \arg\max_{s_k} P(c|s_k)\,P(s_k) \\ &= \arg\max_{s_k}[\log P(c|s_k) + \log P(s_k)] \end{aligned}$$

In our case, we describe the context of *w* in terms of the words v_j that occur in the context.

The Naïve Bayes assumption is that the attributes used for description are all conditionally independent:

$$P(c|s_k) = P(\{v_j|\, v_j \text{ in } c\} \,|\, s_k) = \prod_{v_j \text{ in } c} P(v_j|s_k)$$

With the Naïve Bayes assumption, we get the following modified decision rule for classification:

Decide s' if $s' = \arg\max s_k [\log P(s_k) + \sum_{\text{vj in c}} \log P(v_j|s_k)]$

$P(v_j|s_k)$ and $P(s_k)$ are computed via Maximum-Likelihood estimation, perhaps with appropriate smoothing, from label training corpus:

$$P(v_j|s_k) = \frac{C(v_j,s_k)}{C(s_k)}$$

```
comment: Training
for all senses $s_k$ of w do
    for all words $v_j$ in the vocabulary do
        $P(v_j|s_k) = \frac{C(v_j,s_k)}{C(s_k)}$
    end
end
for all senses $s_k$ of $w$ do
```

```
P ($s_k$) = $\frac{C(s_k)}{C(w)}$
end
comment: Disambiguation
for all senses $s_k$ of $w$ do
    score($s_k$) = log P($s_k$)
    for all word $v_j$ in the context window $c$ do
        score ($s_k$) = score ($s_k$) + log P($v_j$| $s_k$)
    end
end
choose s' = arg max $s_k$ score ($s_k$)[2]
```

Figure 2.Naïve Bayesian Classification for Word Sense Disambiguation

4.3. Sensevl-3 corpus

A corpus is a collection of naturally occurring language text, chosen to characteristics a state or variety of language. Usually a corpus is in machine-readable format and is ideally viewable and analyzable through (a single) software package.

In this system, we use Senseval-3 corpus as training data for Naïve Bayesian classifier. Senseval corpora are common resources for Word Sense Disambiguation. Senseval is a textual corpus in which words are syntactically and semantically tagged. The Senseval-3 corpus consists of approximately 5,000 words of running text from two Wall Street Journal articles and one excerpt from the Brown corpus. It contains a total of 2,212 words tagged with Word Net senses.

4.4. Example

For example, the sentence "He has no ready answer to fit this." First the system tokenized the input sentence and tag the POS for each tokenized word as "He/PP, has/VBZ, no/DT, ready/JJ, answer/NN, to/TO, fit/VB, this/DT. In this example, the target word is 'answer' and other words are bag-of-word. The system then search the word counts file to find probabilities of each sense of target word and bag-of-word. For this example, the target word 'answer' has sense no (3, 5) in the training data. After applying Naïve Bayesian Classification to disambiguate target word, we get 5.555555555555553E-32 for sense no3 and 3.1236984589754263E-14 for sense no 5. The system select the sense no 5 because of highest probabilities. Then, the system access the Word Net for meaning of sense no 5 of 'answer'. The correct meaning for word 'answer' in this sentence is 'a nonverbal reaction'.

5. Proposed System

5.1. System Design

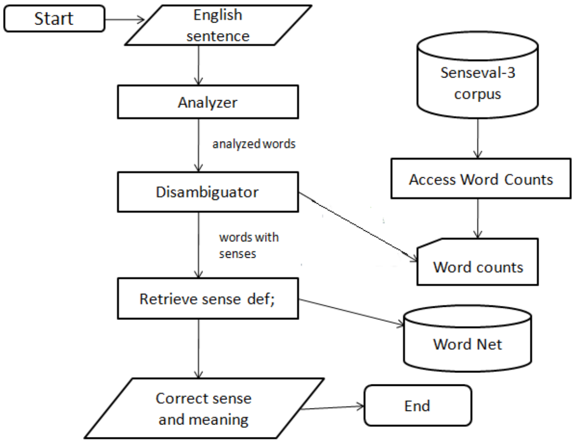

Figure 3.Overview design of the system

The major components of proposed systems are input, Analyzer, Disambiguator, Retrieve sense definition and output.

Firstly, the system takes English Sentence as input. This system use tree tagger as Analyzer. The Analyzer tokenizes the input sentence and tag Part-Of-Speech (POS) for each token. Then the Analyzer passes analyzed words to 'Disambiguator'.

As a preprocessing for Disambiguator, the system access sentence by sentence Seneval-3 corpus to get target words counts and its bag-of-word counts. This system collect counts of words with POS 'Noun' as target word counts and others counts of words as bag-of-word counts in a sentence. Then, these counts are stored in a text file. The Disambiguator takes analyzed words as input and then creates target words list and bag-of-word. This system collect words with POS noun into target words list and other words as bag-of-word.The Disambiguator takes a target word from target words list, collects bag-of-word according to the target word and disambiguate target word by using Naïve Bayesian Classification, and then select the sense with greatest probabilities.

'Retrieve sense definition' component access the Word Net to get the meaning of sense with greatest probabilities.

Finally the system generate the correct sense number and related meaning for each target word in target words list as output.

5.2. Evaluation of System

Accuracy can be measured by sensitivity and specificity. Sensitivity and specificity are the most widely used statistics used to describe a diagnostic test. Sensitivity is referred to as the true positive (recognition) rate (that is, the proportion of positive sample that are correctly identified), while specificity is the true negative rate (that is, the proportion of negative sample that are correctly identified).

$$\text{Sensitivity}=\frac{\text{positive correctly classified}}{\text{total positive}}$$

$$\text{Specificity}=\frac{\text{negative correctly classified}}{\text{total negative}}$$

$$\text{accuracy}=\frac{\text{instances correctly classified}}{\text{total instances}}$$

For evaluation purpose, we group sentences in two groups, first group sentences are composed of words in corpus and second group sentences are composed of words that are not in corpus. There are 60 sentences in first group and 40 sentences in the second group, so there are altogether 100 sentences for evaluation.

The evaluation of the system achieved 82% test accuracy. Figure 4 and 5 are described the comparison of accuracy and error rate and also sensitivity and specificity results of our proposed system respectively.

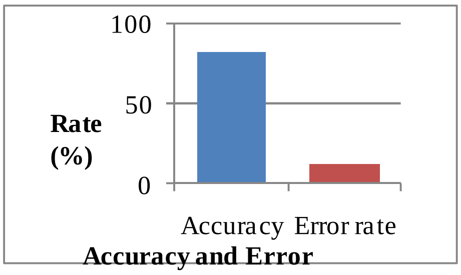

Figure 4. Accuracy and Error Rate

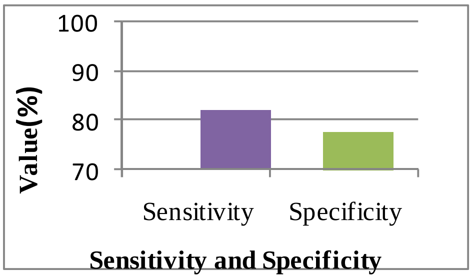

Figure 5. Sensitivity and Specificity Rate

5.3. Limitations

This system has the following limitations:

- This system cannot disambiguate the same target word in the same sentence because of bag-of-word condition.
- This system can disambiguate senses of words which are only in the corpus.
- If probabilities of bag-of-word are same, this system allocates the sense which has greater probabilities to the target word.

6. Conclusions and Future Extensions

6.1. Conclusion

This paper shows that word sense disambiguation can be performed by using Naïve Bayesian Classifier which is based on bag-of-word feature. This approach is evaluated by using nouns words in Senseval-3 corpus, which is extracted from two Wall Street Journal articles and on excerpt from the Brown corpus.

6.2. Future Works

A number of issues have arisen in the course of this work.

Addition to bag-of-word feature, the system can also use either co-occurrence feature or collocation feature, which can disambiguate two same target words in the same sentence.

In this paper, this system use Senseval-3 corpus as training data. We can also use other corpus such as Semcor-3 in which words are already tagged with sense numbers as a training data.

This system disambiguates only words with part of speech ‘Noun’. We can also implement this system for words with other part of speech such as ‘Verb’.

7. References

[1] Escudero G., Marquez L., and Rigau G. 2000a. “*Naive Bayes and Exemplar-Based Approaches to Word Sense Disambiguation Revisited*”. Proceedings of the 14th European Conference on Artificial Intelligence (ECAI), pp. 421-425.

[2] Foundations of Statistical Natural Language Processing

[3] Leacock, C. and Chodorow, M. and Miller, G. 1998. “*Using Corpus Statistics and WordNet Relations for Sense Identification*”. Computational Linguistics, pages 147-165.

[4] Mooney, R. J. 1996. “*Comparative Experiments on Disambiguating Word Senses: An illustration of the role of bias in machine learning*”. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 82-91.

[5] Ng, H.T. and Lee, H.B. 1996. “*Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach*”. Proceedings of the 34th Annual Meeting of the Society for Computational Linguistics (ACL), pp. 40-47.

[6] Pedersen, T. 2000. “*A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation*”. Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL), pp. 63-69.