Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# A Naïve Bayes Approach for Word Sense Disambiguation

**Gurinder Pal Singh Gosal**
Department of Computer Science,
Punjabi University, Patiala, Punjab, India

***Abstract- The word sense disambiguation (WSD) is the task ofautomatically selecting the correct sense given a context and it helps in solving many ambiguity problems inherently existing in all natural languages.Statistical Natural Language Processing (NLP),which is based on probabilistic, stochastic and statistical methods, has been used to solve many NLP problems.The Naive Bayes algorithm which is one of the supervised learning techniques has worked well in many classification problems. In the present work, WSD task to disambiguate the senses of different words from the standard corpora available in the "1998 SENSEVAL Word Sense Disambiguation (WSD) shared task" is performed by applying Naïve Bayes machine learning technique. It is observed that senses of ambiguous word having lesser number of part-of-speeches are disambiguated more correctly. Other key observation is that with lesser number of senses to be disambiguated, the chances of words being disambiguated with correct senses are more.***

***Keywords— Word sense disambiguation, WSD, POS-filtering, ambiguity, Naïve Bayes, supervised learning***

## I. INTRODUCTION

The ambiguity in the senses of the words of different languages does exist inherently in all natural languages used by humans. There are many words in every language which carry more than one meaning for the same word. For example, the word "chair" has one sense which means a piece of furniture and other sense of it means a person chairing say some session. So obviously we need some context to select the correct sense given a situation. Automatically selecting the correct sense given a context is in the core of solving many ambiguity problems. The word sense disambiguation (WSD) is the task to automatically determine which of the senses of an ambiguous (target) word is chosen in the specific use of the word by taking into consideration the context of word's use [1,2].

Having an accurate and reliable word sense disambiguation has been the target of natural language community since long. The motivation and belief behind performing word sense disambiguation is that many tasks which are performed under the umbrella of NLP are highly benefitted with properly disambiguated word senses.Statistical NLP, a special approach of NLP based onthe probabilistic, stochastic and statistical methods, uses machine learning algorithms to solve many NLP problems. AS a branch ofartificial intelligence, machine learning involves computationallylearning patterns from given data, and applying to new or unseen data the pattern which were learned earlier. Machine learning is defined by Tom M.Mitchell as "A computer program is said to learn from experience E with respect to some class of tasksT and performance measure P, if its performance at tasks in T,as measured by P, improves withexperience E [3]."

Learning algorithms can be generally classified into three types: supervised learning, semi-supervised learning and unsupervised learning. Supervised learning technique is based on the idea of studying the features of positive and negative examples over a large collection of annotated corpus. Semi-supervised learning uses both labeled data and unlabeled data for the learning process to reduce the dependence on training data. In the unsupervised learning, decisions are made on the basis of unlabeled data. The methods of unsupervised learning are mostly built upon clustering techniques, similarity based functions and distribution statistics. For automatic WSD,supervised learningis one ofthe most successfulapproaches.

## II. RELATED WORK

When the work started on handling of languages with automatic means, the problem of WSD drew the interest of the researchers at the same time. Therefore, we can say that the WSD task is one of the oldest tasks in computational linguistics.The problem of WSD was introduced to the community by Weaver in 1949 when he presented it as a basic task of MachineTranslation (MT). In his well-known Memorandum on Machine Translation, he stressed that by looking at the context in which the word occurs, this problem of multiple senses of words can be dealt with [4]. The research came out with the importance of immediate context or adjacent words in doing the disambiguation of the senses. The role of the domain in WSD task was also analyzed by Weaver and a lot of work followed in this direction after that generating many specialized dictionaries [5, 6] for sense disambiguation.

There was a view amongst the research community for long that machine translation and word sense disambiguation are tasks have to be dealt independently. WSD was thought to be a very difficult task to achieve given the limited set of resources available at that time.In another study the role of syntactic relations in the task of WSD was discussed by Reifler in his work where he stressed upon the role of grammatical structure [7].

 

In another important work in 1965, Madhu et al.proposed "figure of merit" technique which approached the problem of non-grammatical ambiguity from the viewpoint of probability theory [8]. The technique, in which they calculated sense frequencies using corpora in differentdomains and thereafter used a Bayesian formula to find out theprobability of each sense given the context, was highly effective. By representing the context of words in terms of vectors of co-occurrence and POS features, Pedersen et al. represented another variant technique of WSD which performed well for the task [9].

There are different methodologies which are applied to the task of WSD, such as, knowledge-based methods, supervised methods, unsupervised methods to name a few.Knowledge-based methods primarilydepend uponsome external knowledge bases, such as dictionaries, corpora and lexical semantic resources to tackle theWSD problem. Supervised learning technique, often well suited for target word disambiguation, is based on the idea of studying the features of positive and negative examples over a large collection of manually orautomatically sense-tagged corpora. The main drawback of supervised WSD methods is the needof large amount of training data which is mostly manually annotated and this requires a lot of cost and time investments.

Semi-supervised learning is another machine learning technique which uses both unlabeled data and labeled data for the learning process and thus reduces the reliance on training data. In the unsupervised learning, decisions are made on the basis of unlabeled data. The methods of unsupervised learning are mostly built upon clustering techniques, similarity based functions and distribution statistics. In the WSD task occurrences of words are clustered by applying clustering techniquesbased on their contexts to automatically assign the word senses.In general, machine learning based WSDsystems perform better than other techniques of WSD.

In the category of supervised machine learning algorithms, the classifier based on the Naive Bayes algorithmis one of the techniques which have worked with a great success in many classification problems [10]. This very simplistic but highly effective technique based on Bayesian decision theoryhas done better than many other classification techniques.

The objective of the experiments done in the present workis to disambiguate, analyse and evaluate word sense disambiguation to disambiguate the senses ofdifferent words in a corpus by applying Naïve Bayes machine learning technique used for supervised disambiguation.For the training, testing and gold standard corpora, we used the standard corpora available from the1998 SENSEVAL Word Sense Disambiguation (WSD)shared task [11]. For the implementation of the above machine learning algorithm, customized software was implemented in Java language.

## III. EXPERIMENTATION

### *A. Dataset Source*

For the training, testing and gold standard corpora, we used the standard corpora available from 1998 SENSEVAL Word Sense Disambiguation (WSD) shared task consisting of English-language materials used in the 1998 SENSEVAL Word Sense Disambiguation (WSD) evaluation exercise. Dictionary entries for the 35 selected words, and corpus instances manually sense-tagged according to those dictionary entries, are available.

### *B. Coverage*

Our reference corpus for the experimentation is chosen by selecting 3 words for final task (although the other words were tried during experimentation) from the 35 words used in English SENSEVAL, having their distribution according to part of speech, and the numbers of test instances associated. The 3 words chosen for final experimentation were:

1. accident - (8 senses- all nouns ,267 instances)
2. shake- (8 senses- indeterminate 36 senses, 356 instances)
   **Note:**indeterminate - All test items in files with **-p** suffix; the part of speech of the word to be disambiguated has not been predetermined.
3. behaviour – (3 senses – all nouns, 279 instances)

The word “accident” was chosen as it was the first word in the list and further it was just having all the 8 senses as nouns. The word “shake” was chosen as it was having maximum number of senses (36) out of all the 35 words. The word “behaviour” was chosen as a special case as it was found during experimentation on different words in the set that it was being disambiguated 100%.

The training data for these words “accident”, “shake” and “behaviour” are available as files “accident.cor”, “behaviour.cor”, “shake.cor” in the folder in the project “.\snlp\assignment2data\train”.

The test data for these words “accident”, “shake” and “behaviour” are available as files “accident-n.eval”, “behaviour-n.eval”, “shake-p.eval” in the folder in the project “.\snlp\assignment2data\test”.

The gold data for these words “accident”, “shake” and “behaviour” are available as files “accident.n”, “behaviour.n”, “shake.p” in the folder in the project “.\snlp\assignment2data\gold”.

***Word*:**For our purposes the word is considered as a word token without punctuation marks and other unwarranted symbols such as, :-).

### *C. Software*

The word sense disambiguation tool is implemented in Java language.The algorithm is based on the “Naïve Bayes” machine learning technique. For word sense disambiguation Bayes classifier is based on the idea that it looks at the words around the ambiguous word in a large context window. The classifier combines the evidence from all features.

 

The implemented code for applying this technique performs certain functions implemented in terms of several routines that include:

- Routine for getting text from dataset (from training data for training phase, from test data for testing purpose).
- Routine for extracting words as tokens in raw form.
- Routinefor refining the words with refinement rules.
- Routine to deal with window sizesof context words, stop word filtering of context words.
- Routine forfrequency engine for frequency count of words, context words to get vocabulary set and size needed in the calculations
- Routine for calculating required probabilities needed in the Naïve Bayes algorithm.
- Routine for disambiguating the word senses based on all the available inputs provided by the various routines above and outputting the results in the desired format.

### *D. Context Window Size*

The feature vectors are chosen for the target word by taking a context window of n word around the target word. As it is very easy to change the size of window for context words as features, you can experiment with the different sizes of window by just changing the size in the parameter.

For our experimentation we used two window sizes 3 and 5 (although it was experimented for sizes of 2 to 8) and these are shown for the evaluation purposes to show the effect of size.

### *E. Smoothing*

To smooth the probability distribution for sparse set of data over large vocabularies, the process of giving a little bit of probability space to unseen events is applied known as smoothing or discounting. Out of the many smoothing techniques, we used smoothing technique of Add-delta smoothing (Lidstone's law), also known as the Expected Likelihood Estimation (ELE) or the Jeffreys-Perks Law.

The code in the software implements the application of this smoothing technique.

### *F. Stop Word Filtering*

In one set of the experimentation, the feature vectors are chosen by taking a window of n word around the target word by not filtering the stop words.

The other set of experimentation is done by filtering the stop words in the feature vector and for list of stop words which wegot from 2 sources:

- NLTK's stop word list of English language- 127 words
- Consolidated list from other sources (also includes these 127 words) – 332 words

### *G. POS Filtering Tagger (If needed for future experimentation, see Future Tasks)*

To do POS Filtering for context words to be included in feature vectors (seen as a future task), aJava routine is included in the code for POS filtering of words by using Stanford Log-linear Part-Of-Speech Tagger [12, 13]. In this application, the *english-left3words-distsim.tagger model*has been used. It's nearly as accurate (96.97% accuracy vs. 97.32% on the standard WSJ22-24 test set) and is an order of magnitude faster. The English taggers use the Penn Treebank tag set [14].

## IV. RESULTS AND DISCUSSIONS

We will first look at the results obtained and then will analyze subsequently.

### *A. Disambiguation of words based on Naive Bayes (Without Stop Word Filtering)*

The first set of experiments explained here is trying to disambiguate the senses of the 3 words whenthe context words are not being filtered for stop words. The context word window size as explained earlier can be set for various sizes by passing parameter value. Here we analyze with window size 3 and window size 5.

Table 1 Disambiguation based on context window sizes 3 & 5 and without filtering stop words

| **Ambiguous Words** | **No of Senses of the word and types of POS** | **No of instances to be tested** | **Disambiguated correctly (Window Size 3)** | **Disambiguated correctly (Window Size 5)** | **Senses used in training** |
|---|---|---|---|---|---|
| ***accident*** | 8 senses / all nouns | 267 | 206 (77.15%) | 205 (76.80%) | 1303 |
| ***behaviour*** | 3 senses / all nouns | 279 | 279 (100%) | 278 (99.64%) | 1003 |
| ***shake*** | 36 senses indeterminate | 356 | 180 (50.5%) | 184 (51.70%) | 1024 |

 

### *1) Context Window Size of 3 and 5*

For words *accident, behaviour*and *shake* using the window sizes 3 and 5 and without filtering stop words, the outcome obtained on running the algorithm is shown in Table 1.

### *B. Disambiguation of words based on Naive Bayes (With Stop Word Filtering)*

The other set of experiments is trying to disambiguate the senses of the 3 words when the context words are filtered for stop words. The context word window size as explained earlier can be set for various sizes by passing parameter value. Here we analyze with window size 3 and window size 5 with stop word filtering.

Table 2 Disambiguation based on context window size 3 and 5 and with filtering stop words

| Ambiguous Words | No of Senses of the word and types of POS | No of instances to be tested | Disambiguated correctly (Window Size 3) | Disambiguated correctly (Window Size 5) | Senses used in training |
|---|---|---|---|---|---|
| ***accident*** | 8 senses / all nouns | 267 | 206 (77.15%) | 206 (77.15%) | 1303 |
| ***behaviour*** | 3 senses / all nouns | 279 | 279 (100%) | 279 (100%) | 1003 |
| ***shake*** | 36 senses indeterminate | 356 | 170 (51.70%) | 184 (51.70%) | 1024 |

Table 3 shows the effect of window sizes and stop word filtering on the feature vector by taking some excerpts of data.

Table 3Excerpts showing the effect of window sizes and stop word filtering on the feature vector

| Line No | Context words with Window Size 3 (Without stop word Filtering) | Context words with Window Size 5 (Without stop word Filtering) |
|---|---|---|
| 700275 | [an, a, in, stranger, that, can] | [an, a, in, stranger, that, can, vital, release, is, a] |
| 700125 | [an, and, maintain, emergency, to, service] | [an, and, maintain, emergency, to, service, guarantee, even, a, if] |
| 700274 | [no, he, course, said, of, out] | [no, he, course, said, of, out, is, loud, It, testing] |
| 700124 | [fatal, involving, 14, small, been, planes] | [fatal, involving, 14, small, been, planes, have, and, There, helicopters] |
| 700273 | [An, in, France, would] | [An, in, France, would, no, have] |
| 700123 | [her, we, after, went, day, to] | [her, we, after, went, day, to, The, the, swimming] |
| 700122 | [the, when, in, an, injured, Army] | [the, when, in, an, injured, Army, were, ambulance, Luke, hit] |
| 700279 | [of, in, number, 1983, total, was] | [of, in, number, 1983, total, was, The, 1255, but] |
| 700121 | [the, say, of, the, cause, dead] | [the, say, of, the, cause, dead, the, man, investigating, was] |
| 700278 | [future, prevention, promote, but, would, reduce] | [future, prevention, promote, but, would, reduce, tactic, the, This, incentive] |

| Line No | Context words with Window Size 3 (With stop word Filtering) | Context words with Window Size 5 (With stop word Filtering) |
|---|---|---|
| 700275 | [stranger] | [stranger, vital, release] |
| 700125 | [maintain, emergency, service] | [maintain, emergency, service, guarantee, even] |
| 700274 | [course, said] | [course, said, loud, testing] |
| 700124 | [fatal, involving, 14, small, planes] | [fatal, involving, 14, small, planes, helicopters] |
| 700273 | [France, would] | [France, would] |
| 700123 | [went, day] | [went, day, swimming] |
| 700122 | [injured, Army] | [injured, Army, ambulance, Luke, hit] |
| 700279 | [number, 1983, total] | [number, 1983, total, 1255] |
| 700121 | [say, cause, dead] | [say, cause, dead, man, investigating] |
| 700278 | [future, prevention, promote, would, reduce] | [future, prevention, promote, would, reduce, tactic, incentive] |

 

## V. CONCLUSION

The following are some important observations in this task of using "Naïve Bayes" approach for word sense disambiguation on the subset of standard corpora available in the "1998 SENSEVAL Word Sense Disambiguation (WSD) shared task":

1. As we see the accuracy of accident with all senses of nouns, it can be said from the set of experimentation that senses of ambiguous word having lesser number of part-of-speech are disambiguated more correctly. Nouns tend to have more accuracy than other part-of-speeches, e.g. "behaviour", "accident".
2. The second observation is that with lesser number of senses to be disambiguated the chances of it being disambiguated correctly are more, e.g. "behaviour" in our experimentation.
3. By increasing the size of context word window it marginally affects the results, most of the times increasing accuracy but in one case it reduced by 1 instance. As the word sense disambiguation on basis of Naive Bayes classifier, uses the words in context window at the cost of a somewhat unrealistic independence assumption (bag of words model) and ignoring structure and linear ordering of words, so it is somewhat understandable. But the word of caution is that this observation is again based on very limited set of experimentation here.
4. By applying stop word filtering has surprisingly not affected the results at all in our set of experimentation as it was expected to increase accuracy.
5. The size of the training corpus affects the results although not to very large extent in our set of experimentation. For example, reducing by 50% of corpus for *shake* reduced accuracy by 4%. (170 correct instances instead of 184)

## VI. FUTURE WORK

It will be interesting to observe the effects of POS tagging and filtering of context words to be used in feature vectorsfor the future tasks. The code can be amended to analyze and evaluate based on this filtering.Alsothe gold set of corpora contains more than one sense being disambiguated for many words; this motivates to look for doing the same with the automated disambiguation when it meets certain criteria of selection. It may require some tweaking in code to return 2 senses when the score are within the selected range.

As Naive Bayesian classifier is the only technique used in the experiments, it would be logical and not very difficult to use the same feature vectors with other machine learning techniques, such as, decision trees, neural networks and then compare and evaluate the results. Further it will be interesting to run the program on whole set of 35 words and average the results for better evaluation.

## REFERENCES

[1] Foundations of Statistical Natural Language Processing, by Chris Manning and HinrichSchutze, MIT Press, 1999.
[2] Speech and Language Processing, Daniel Jurafsky& James H. Martin. Prentice Hall, 2nd edition, 2008.
[3] Mitchell, T. M. (1997). Machine learning. WCB.
[4] Weaver, W. (1955). Translation. *Machine translation of languages*, *14*, 15-23.
[5] Oswald Jr, V. A. (1957). The rationale of the idioglossary technique. *Research in Machine Translation, Georgetown University Press, Washington, DC*, 63-69.
[6] Hutchins, W. J. (2004). The Georgetown-IBM experiment demonstrated in January 1954. In *Machine Translation: From Real Users to Research* (pp. 102-114). Springer Berlin Heidelberg.
[7] Reifler, E. (2003). The mechanical determination of meaning. *Readings in machine translation*, 21-36.
[8] Madhu, S., & Lytle, D. W. (1965). A figure of merit technique for the resolution of non-grammatical ambiguity. *Mechanical translation*, *8*(2), 9-13.
[9] Pedersen, T., & Bruce, R. (1997). Distinguishing word senses in untagged text. *arXiv preprint cmp-lg/9706008*.
[10] Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46). IBM New York.
[11] Edmonds, Phillip. SENSEVAL: The evaluation of word sense disambiguation systems, Published in the ELRA Newsletter, Vol. 7 No. 3, 2002.
[12] Kristina Toutanova and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp. 63-70.
[13] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.
[14] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz: Building a Large Annotated Corpus of English: The Penn Treebank, *in* Computational Linguistics, Volume 19, Number 2 (June 1993), pp. 313—330

 