Project Report
Word sense Disambiguation
Implementation of Different Methods

Anant Rajendra Dashpute
(SC20M144)

Abstract

Word Sense Disambiguation (WSD) is an important but challenging technique in the area of natural language processing (NLP). Hundreds of WSD algorithms and systems are available. WSD is typically configured as an intermediate task, either as a stand-alone module or properly integrated into an language application. However, the success of WSD in real-world applications are still not desirable yet. So To get deep understanding, how to approach this problem ?, I have implemented few famous algorithms. Some of them are improved using new Natural language preprocessing techniques (POS tag, word Stem, String Tokenizer). This report will summarize Four popular methods in the area of Knowledge-based and Supervised learning.

The increasing availability of wide-coverage, rich lexical knowledge resources, as well as the construction of large-scale coarse-grained sense inventories, seems to open new opportunities for disambiguation approaches, especially when aiming at semantically enabling applications in the area of human-language technology.

Contents

1 Introduction

1.1 Word sense disambiguation

Word sense disambiguation (WSD) is a crucial task in the area of NLP. WSD is a natural classification problem: If you are provided with a word and all of its possible meanings as present in the dictionary then classification of an presence of the word in context into one or more of its meaning classes. WSD has impact on many Natural Language Processing applications like Information Retrieval, Information Extraction and Machine Translation. Word sense disambiguation (WSD) is a problem in which it is hard to determine which ”sense”or which meaning of some word is initiated when we make use of that word somewhere in particular with a predetermined context. One of the central and most widely investigated problems in NLP is word sense disambiguation. In WSD, given a sentence aligned from corpus and give different senses of words present in that sentence. The senses are taking from Word-net, somewhere these senses are noun, some where these are adjective, some where these are verb and adverb etc.

A lot of techniques have been researched, including dictionary-based methods that use the knowledge encoded in lexical resources, supervised machine learning methods in which a classifier is trained for each distinct word on a corpus of manually sense-annotated examples, completely unsupervised methods that cluster occurrences of words, thereby inducing word senses. Among these, supervised learning approaches have been the most successful algorithms to date.

1.2 History

The task of WSD is a historical one in the field of Natural Language Processing (NLP). WSD was first formulated into as a distinct computational task during the early days of machine translation in the 1940s, making it one of the oldest problems in computational linguistics.

In the 1970s, WSD was a subtask of semantic interpretation systems developed within the field of artificial intelligence, starting with Wilks’ preference semantics. However, since WSD systems were at the time largely rule-based and hand-coded they were prone to a knowledge acquisition bottleneck.

By the 1980s large-scale lexical resources, such as the Oxford Advanced Learner’s Dictionary of Current English (OALD), became available: hand-coding was replaced with knowledge automatically extracted from these resources, but disambiguation was still knowledge-based or dictionary-based.

In the 1990s, the statistical revolution advanced computational linguistics, and WSD became a paradigm problem on which to apply supervised machine learning techniques.

The 2000s saw supervised techniques reach a plateau in accuracy, and so attention has shifted to coarser-grained senses, domain adaptation, semi-supervised and unsupervised corpus-based systems, combinations of different methods, and the return of knowledge-based systems via graph-based methods. Still, supervised systems continue to perform best.

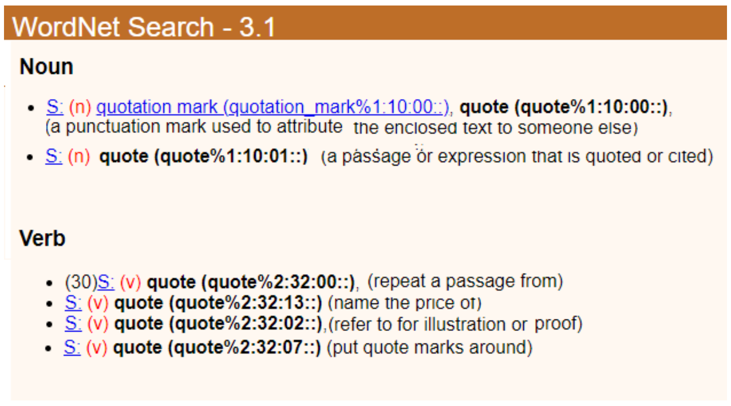
2 Dictionary

2.1 WordNet

WordNet is a lexical database of semantic relations between words in more than 200 languages. WordNet links words into semantic relations including synonyms, hyponyms, and meronyms. The synonyms are grouped into synsets with short definitions and usage examples.

WordNet includes the lexical categories nouns, verbs, adjectives and adverbs but ignores prepositions, determiners and other function words.

Words from the same lexical category that are roughly synonymous are grouped into synsets. The different senses of a polysemous word form are assigned to different synsets. The meaning of a synset is further clarified with a short defining gloss and one or more usage examples.
An example synset is:



WordNet Search - 3.1

Noun

- S: (n) quotation mark (quotation_mark%1:10:00::), **quote (quote%1:10:00::)**, (a punctuation mark used to attribute the enclosed text to someone else)
- S: (n) **quote (quote%1:10:01::)** (a passage or expression that is quoted or cited)

Verb

- (30)S: (v) **quote (quote%2:32:00::)**, (repeat a passage from)
- S: (v) **quote (quote%2:32:13::)** (name the price of)
- S: (v) **quote (quote%2:32:02::)**,(refer to for illustration or proof)
- S: (v) **quote (quote%2:32:07::)** (put quote marks around)

Figure 2.1: Word-net search by Princeton university

All synsets are connected to other synsets by means of semantic relations. These relations, which are not all shared by all lexical categories, include:

1. hypernyms: Y is a hypernym of X if every X is a (kind of) Y (canine is a hypernym of dog)

2. hyponyms: Y is a hyponym of X if every Y is a (kind of) X (dog is a hyponym of canine)

3. meronym: Y is a meronym of X if Y is a part of X (window is a meronym of building)

4. holonym: Y is a holonym of X if X is a part of Y (building is a holonym of window)

5. hypernym: the verb Y is a hypernym of the verb X if the activity X is a (kind of) Y (to perceive is an hypernym of to listen)

6. troponym: the verb Y is a troponym of the verb X if the activity Y is doing X in some manner (to lisp is a troponym of to talk)

These semantic relations hold among all members of the linked synsets. Individual synset members (words) can also be connected with lexical relations.

3 NLP Techniques

3.1 Tokenization

Tokenization is the process of demarcating and possibly classifying sections of a string of input characters. The resulting tokens are then passed on to some other form of processing. The process can be considered a sub-task of parsing input.

```
import nltk
from nltk.tokenize import word_tokenize
input='The quick brown fox jumps over the lazy dog'
tokens= tokenizer.tokenize(input)
print(tokens)

```

```
output-
[(The)  (quick)  (brown)  (fox)  (jumps)  (over)  (the)  (lazy)  (dog)]

```

3.2 POS Tagging

Part-of-speech tagging (POS tagging) is the task of tagging a word in a text with its part of speech. A part of speech is a category of words with similar grammatical properties. Common English parts of speech are noun, verb, adjective, adverb, pronoun, preposition, conjunction, etc.

```
def change_pos(i):
if i=='VB' or i=='VBD' or i=='VBG' or i=='VBN' or i=='VBP' or i=='VBZ' :
return 'v'
elif i== 'NN' or i=='NNS' or i=='NNP' or i=='NNPS' :
return 'n'
elif i== 'RB'or i=='RBR' or i=='RBS' or i=='RP':
return 'r'
elif i=='JJ'or i=='JJR'or i=='JJS':
return 's'
else:
return 'a'

```

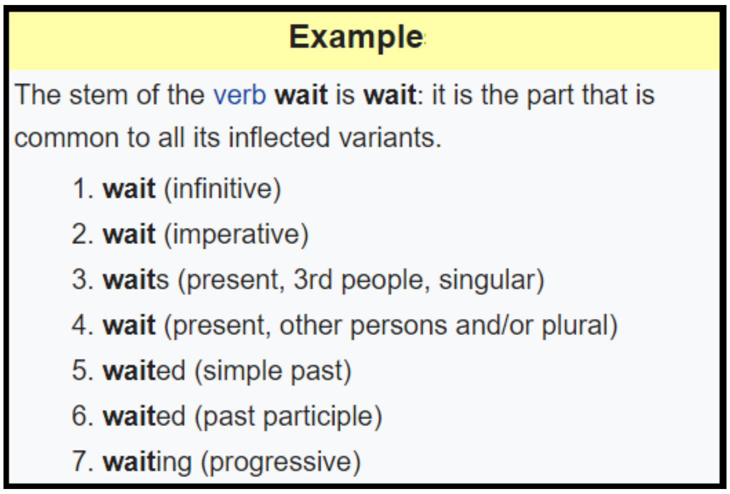
As wordnet provide senses for few pos tags, so we are updating pos tags irrespective of their tense, Using change-pos function.

```
import numpy as np
import nltk
from nltk.tokenize import word_tokenize
input = 'She chopped the vegetables with a c h e f s  knife'
token = nltk.word_tokenize(input)
tags = nltk.pos_tag(token)
tags=np.array(tags)
for i in range(len(tags)):
tags[i][1]=str(change(tags[i][1]))
print(tags)

```

```
output-
[['She' 'a'] ['chopped' 'v'] ['the' 'a'] ['vegetables' 'n'] ['with' 'a'] ['
  chef' 'n'] ['s' 'v'] ['knife' 'n'] ]

```

3.3 Word stemming

In linguistics, a stem is a part of a word responsible for its lexical meaning. The term is used with slightly different meanings and would depend on the morphology of the language in question. There can be different variants for a word with respect to tenses,and its use.



Example

The stem of the verb **wait** is **wait**: it is the part that is common to all its inflected variants.

1. **wait** (infinitive)
2. **wait** (imperative)
3. **wait**s (present, 3rd people, singular)
4. **wait** (present, other persons and/or plural)
5. **wait**ed (simple past)
6. **wait**ed (past participle)
7. **wait**ing (progressive)

What is Root Word?

A root (or root word) is the core of a word that is irreducible into more meaningful elements.In morphology, a root is a morphologically simple unit which can be left bare or to which a prefix or a suffix can attach.

The root word is the primary lexical unit of a word, and of a word family (this root is then called the base word), which carries aspects of semantic content and cannot be reduced into smaller constituents.

Content words in nearly all languages contain, and may consist only of, root morphemes. However, sometimes the term ”root” is also used to describe the word without its inflectional endings, but with its lexical endings in place.

Stemming

Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word.

```python
import numpy as np
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
ps=PorterStemmer()
def stem(i):
return ps.stem(i)

```

```python
input ='She chopped the vegetables with a c h e f s knife'
tokens = word_tokenize(input1)
for i in range(len(tokens)):
tokens[i]=str(stem(tokens[i]))
print(tokens)

```

```
output-
['she', 'chop', 'the', 'veget', 'with', 'a', 'chef', 's', 'knife']

```

4 Approaches and methods

There are many approaches includes :

1. **Knowledge-based Methods**
2. **Machine learning Methods**
3. **Deep Neural Network Methods**

There are four conventional approaches to WSD:

1. **Dictionary- and knowledge-based methods** These rely primarily on dictionaries, thesauri, and lexical knowledge bases, without using any corpus evidence.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.

2. **Supervised methods**
   Supervised methods are based on the assumption that the context can provide enough evidence on its own to disambiguate words (hence, common sense and reasoning are deemed unnecessary).Supervised methods are subject to a new knowledge acquisition bottleneck since they rely on substantial amounts of manually sense-tagged corpora for training, which are laborious and expensive to create.

3. **Unsupervised methods**
   completely external information and work directly from raw not annotated corpora. These methods are also known under the name of word sense discrimination.

4. **Semi-supervised methods**
   Because of the lack of training data, many word sense disambiguation algorithms use semi-supervised learning, which allows both labeled and unlabeled data. The Yarowsky algorithm was an early example of such an algorithm. It uses the 'One sense per collocation' and the 'One sense per discourse' properties of human languages for word sense disambiguation. From observation, words tend to exhibit only one sense in most given discourse and in a given collocation.

Almost all these approaches work by defining a window of n content words around each word to be disambiguated in the corpus, and statistically analyzing those n surrounding words. Two shallow approaches used to train and then disambiguate are Naïve Bayes classifiers and decision trees. In recent research, kernel-based methods such as support vector machines have shown superior performance in supervised learning.

4.1 Dictionary- and knowledge-based methods

The Lesk algorithmis the seminal dictionary-based method. It is based on the hypothesis that words used together in 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 disambiguted by finding the pair of dictionary senses with the greatest word overlap in their dictionary definitions.

4.1.1 Lesk Algorithm

The lesk algorithm is a traditional and very classical algorithm for word sense disambiguation. It was presented by Michael E. Lesk in 1986.

It recognizes word meanings in a particular aspect by bringing in use overlapping between definitions providing approximately 50-70 % accuracy. The original Lesk algorithm basically works on the principle of disambiguation of a mark word by contrasting its meanings set with the meaning sets of neighboring words. That sense is assigned to the mark word whose glossary has the most covering words with the glossaries of its surrounding words.

Algorithm finds meaning of each mark word which has maximum closeness to mark sense. Relatedness scores are calculated between the target sense and each most related context sense. Once the algorithm finds a value for all senses of mark word, the sense with greatest overlap value is designated to mark word.

```
function Simplified_Lesk(word, sentence)
  best-sense←most frequent sense for word
  max-overlap←0
   context←set of words in sentence
   for each sense in senses of word do
     signature←set of words in the gloss_examples of sense
       overlap←Compute_overlap(signature, context)
       if overlap > max-overlap then
                 max-overlap←overlap
                 best-sense←sense
   end
   return(best-sense){ returns best sense of word}
```

The greatest deficiency of straightforward lesk calculation is that lexicon definitions are regularly short and simply don't have enough words for this calculation then it can malfunction. so to solve this issue we try attaching this algorithm to Word-Net properties. Word-net is semantically organized. Issue of short definitions is can be solved by adding more features among meanings of word which have close relatedness with them in Word-Net.

Overlap of Sense Definitions

A simple and intuitive knowledge-based approach relies on the calculation of the word overlap between the sense definitions of two or more target words. This approach is named gloss overlap or the Lesk algorithm. Given a two word context (w1, w2), the senses of the target words whose definitions have the highest overlap (i.e., words in common) are assumed to be the correct ones. Formally, given two words w1 and w2, the following score is computed for each pair of word senses Senses(w1) and Senses(w2):

$$score_{Lesk}(S_1, S_2) = | gloss(S_1) \cap gloss(S_2) |,$$

where gloss(Si) is the bag of words in the textual definition of sense Si of wi. The senses which maximize the above formula are assigned to the respective words. However, this requires the calculation of gloss overlaps.

Given the exponential number of steps required, a variant of the Lesk algorithm is currently employed which identifies the sense of a word w whose textual definition has the highest overlap with the words in the context of w. Formally, given a target word w, the following score is computed for each sense S of w:

$$score_{LeskVar}(S) = \mid context(w) \cap gloss(S) \mid ,$$

4.1.2 Updated Lesk Algorithm

Apply Lesk's fundamental method to make advantage of synonyms having strong interconnections between them offered by Word-Net.

In order to get more accuracy a refined technique can be implemented to beat the limitations of lesk algorithm. I have to find the correct meaning of every word according to the context in a sentence then each word whose meaning has to identified is taken as feature for bag of word. creating a bag of word for each sense for word using properties of wordnet (like examples, hypernyms, meronyms, and there examples etc.). Also to get better accuracy i will be removing all the stopwords and make all the words in bag of words in stem word format.

Pseudo-Algorithm is summarized in steps described below:

```
Result: Best sense
BestSense = most Frequent sense ;
MaxOverlap = 0 ;
tag= getPOS(word,sentence);
getPOS find pos for word argument with resprct to sentence
context = set of tokens of input sentence;
NewContext = getStem(context);
getStem function takes tokens and returns root word of each token
Senses = wordnet.synset(word);
Filtering senses with respect to pos tag of ambiguous word
NewSenses = Sense with same tags;
for Every sense in NewSenses do
    gloss= bag of words of each sense example from wordnet;
    overlap= compute overlap (NewContext , gloss);
    if $overlap \geq MaxOverlap$ then
        MaxOverlap = overlap;
        BestSense = sense;
    end
end
```

Algorithm 1: Updated Lesk Algorithm

WordNet as a Graph

WordNet is a lexical knowledge base for English that defines words, meanings, and relations between them. The basic unit in WordNet is a synset, which is a set of synonyms and represents a concept. WordNet defines several semantic relations between synsets, including ISA relations (hypernym/hyponym), PART-OF relations (meronym/holonym) and others.

To represent WordNet as a graph,They use an instance-centric data representation, which defines synsets as vertices, and relations or sets of relations as links. The graph can be constructed as an undirected graph, with no orientation defined for edges, or as a directed graph, in which case a direction is arbitrarily established for each relation (e.g. hyponym hypernym).Given a subset of the WordNet synsets, as identified in a given text or by other selectional criteria, and given a semantic relation, a network is constructed by identifying all the synsets (vertices) in the given subset that can be linked by the given relation (edges). Relations can be also combined, for instance a graph can be constructed so that it accounts for both the ISA and the PART-OF relations between the vertices in the graph.

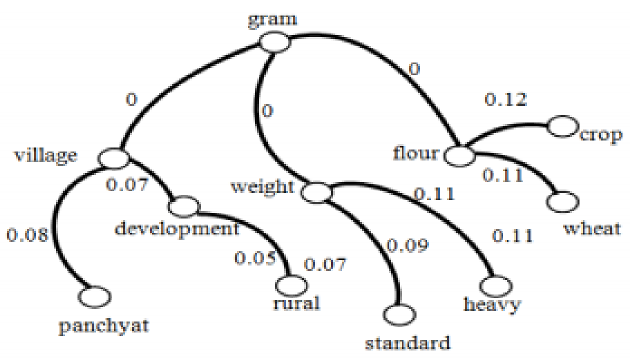
4.1.1 Page rank Algorithm

Google's PageRank link-analysis algorithm (Brin and Page, 1998) have been used for analyzing the link-structure of the World Wide Web to provide global, content independent ranking of Web pages. PageRank can be singled out as a key element of the paradigm-shift Google has triggered in the field of Web search technology, by providing a Web page ranking mechanism that relies on the collective knowledge of Web architects rather than content analysis of individual Web pages.

In short, PageRank is a way of deciding on the importance of a vertex within a graph, by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information. Using same approach to lexical and semantic knowledge graphs like WordNet (Miller, 1995) for language processing applications, where knowledge drawn from an entire text can be used in making local ranking/selection decisions.

Given an input piece of text (typically one sentence, or a small set of contiguous sentences), we want to disambiguate all open-class words in the input taken the rest as context. In this framework, we need to rank the senses of the target words according to all other words in the context.

Supposing that each first-order classifier provides a ranking of the senses for a given target word w, the rank-based combination consists in choosing that sense of w which maximizes the sum of its ranks output by the systems PR= C1,C2, ... , Cm
(we take ranks so that the best ranking sense provides the highest contribution):



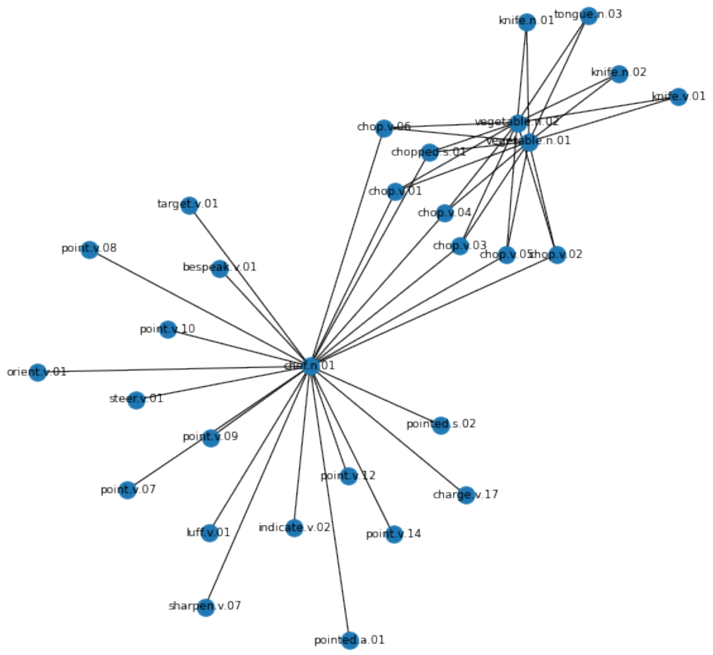
The combination of PageRank with the WordNet sense frequency information is done in following steps:

1. get WordNet frequency ordering by removing the random permutation of senses
2. Link weight is equal to overlap of words in senses of consecutive words

While a simple product of the two ranks already provides an improvement over both algorithms the following formula which prioritizes the first sense provides the best results:
Rank = 4 FR PR where,
FR represents- the WordNet sense frequency,
PR represents the rank computed by PageRank,

Results

Example Input
:-**'She chopped the vegetables with a chef's pointed knife'**
After performing Page rank algorithm:



Using rank calculations following sense are allocated to contexts

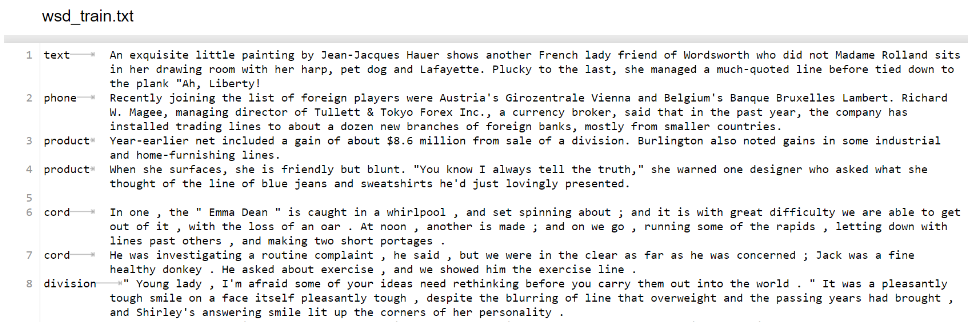
```
chopped : form or shape by chopping
vegetables : edible seeds or roots or stems or leaves or bulbs or tubers or
  non sweet fruits of any of numerous herbaceous plant
knife : edge tool used as a cutting instrument; has a pointed blade with a
  sharp edge and a handle
chef : a professional cook
pointed : indicate a place, direction, person, or thing; either spatially or
     figuratively

```

4.2 Probabilistic Approach- Supervised Learning

Supervised learning need suitable data set for training and testing purposes.

4.2.1 Dataset

As it is supervised learning, data is with lable. here Data is in text format having two features. word as a label and sentence given with respect to that word. The data-set contains two files for training and testing model.



wsd_train.txt

```
text	An exquisite little painting by Jean-Jacques Hauer shows another French lady friend of Wordsworth who did not Madame Rolland sits
	in her drawing room with her harp, pet dog and Lafayette. Plucky to the last, she managed a much-quoted line before tied down to
	the plank "Ah, Liberty!
phone	Recently joining the list of foreign players were Austria's Girozentrale Vienna and Belgium's Banque Bruxelles Lambert. Richard
	W. Magee, managing director of Tullett & Tokyo Forex Inc., a currency broker, said that in the past year, the company has
	installed trading lines to about a dozen new branches of foreign banks, mostly from smaller countries.
product	Year-earlier net included a gain of about $8.6 million from sale of a division. Burlington also noted gains in some industrial
	and home-furnishing lines.
product	When she surfaces, she is friendly but blunt. "You know I always tell the truth," she warned one designer who asked what she
	thought of the line of blue jeans and sweatshirts he'd just lovingly presented.

cord	In one , the " Emma Dean " is caught in a whirlpool , and set spinning about ; and it is with great difficulty we are able to get
	out of it , with the loss of an oar . At noon , another is made ; and on we go , running some of the rapids , letting down with
	lines past others , and making two short portages .
cord	He was investigating a routine complaint , he said , but we were in the clear as far as he was concerned ; Jack was a fine
	healthy donkey . He asked about exercise , and we showed him the exercise line .
division	" Young lady , I'm afraid some of your ideas need rethinking before you carry them out into the world . " It was a pleasantly
	tough smile on a face itself pleasantly tough , despite the blurring of line that overweight and the passing years had brought ,
	and Shirley's answering smile lit up the corners of her personality .
```

4.2.2 Naive Bayes Algorithm

In statistics, naive Bayes classifiers are a family of simple ”probabilistic classifiers” based on applying Bayes’ theorem with strong (naïve) independence assumptions between the features.

A Naive Bayes classifier is a simple probabilistic classifier based on the application of Bayes’ theorem. It relies on the calculation of the conditional probability of each sense Si of a word w given the features fj in the context. The sense S which maximizes the following formula is chosen as the most appropriate sense in context:

Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

Probabilistic model

Naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector

$$
\begin{aligned}
p(S_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, S_k) \\
&= p(x_1 \mid x_2, \dots, x_n, S_k)\ p(x_2, \dots, x_n, S_k) \\
&= p(x_1 \mid x_2, \dots, x_n, S_k)\ p(x_2 \mid x_3, \dots, x_n, S_k)\ p(x_3, \dots, x_n, S_k) \\
&= \cdots \\
&= p(x_1 \mid x_2, \dots, x_n, S_k)\ p(x_2 \mid x_3, \dots, x_n, S_k) \cdots\ p(x_n \mid S_k)\ p(S_k)
\end{aligned}
$$

So by joint probability the formula becomes,

$$
\begin{aligned}
p(C_k \mid x_1, \dots, x_n) &\propto p(S_k, x_1, \dots, x_n) \\
&\propto p(S_k)\ p(x_1 \mid S_k)\ p(x_2 \mid S_k)\ p(x_3 \mid S_k)\ \cdots \\
&\propto p(S_k) \prod_{i=1}^{n} p(x_i \mid S_k)\,,
\end{aligned}
$$

From among all senses of an ambiguous word, the classifier assigns that sense which maximizes the probability of that sense given the feature set of the target word. Mathematically,

$$\mathbf{S} = \boldsymbol{argmax}_{s \in senses} P(s|V_W)$$

$$= \boldsymbol{argmax}_{s \in senses} P(V_W|s)P(s)$$

where is the feature vector. It contains all the context words in stem format, (bag of words), Co-occurrence vector (number of times a word occur in the bag of words around it). Equation 2 is obtained by applying Bayes rule and considering naïve independence assumption which states that each feature is independent of all other features in the feature vector.

Naive Bayes result

WSD train data:
for class cord —- 2357 No. of words present in its bag of words
for class division —- 2637 No. of words present in its bag of words
for class phone —- 2544 No. of words present in its bag of words
for class text —- 3123 No. of words present in its bag of words
for class formation —- 2805 No. of words present in its bag of words

Prior Probability
'cord': 0.19749447310243184,
'division': 0.184966838614591,
'phone': 0.2247605011053795,
'text': 0.20854826823876196,
'formation':0.18422991893883567

After testing Predicted word sense with 75.27272727272727% accuracy.

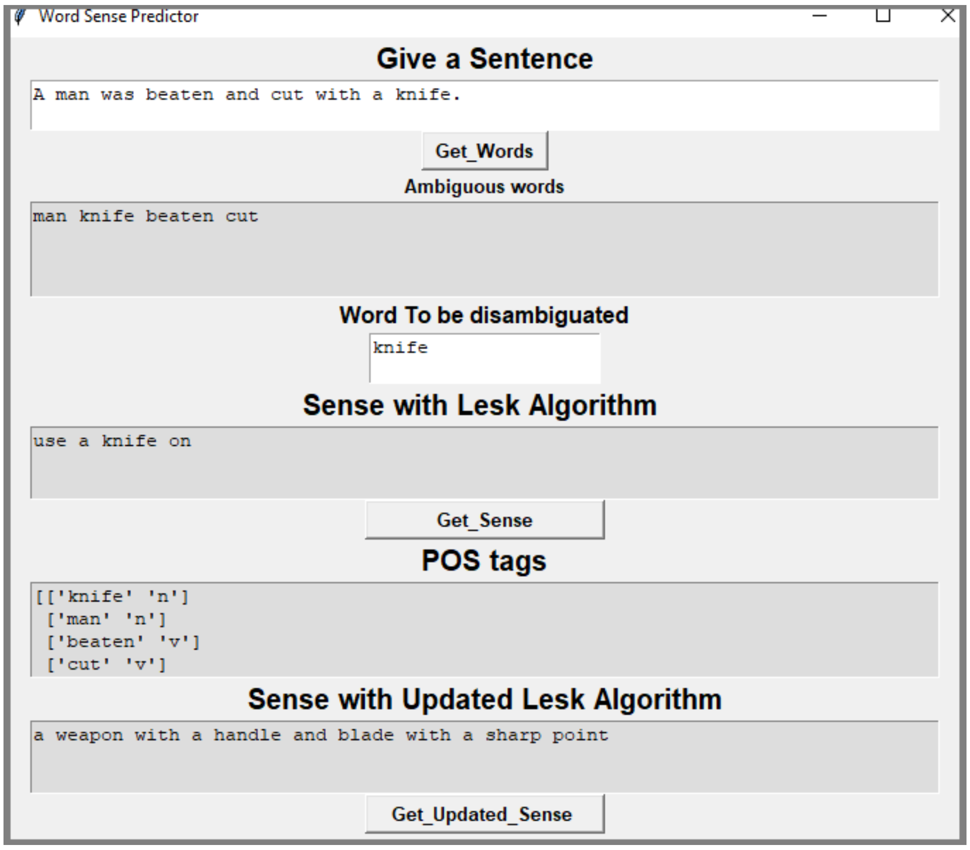
5 Conclusion

1. For Word sense Disambiguation, Word-Net(semantic network) plays very important role, which concludes that NLp tasks like WSD require base dictionaries to perform better.
2. POS tags is worth it (when used as a filter, as it reduces number of senses to consider). It also use to get proper sense as pos tags are allotted based on structure of input sentences.
3. Word stem technique improves accuracy way more than expexcted. Each WSD algorithm performs better when we take stemming operation in consideration.

5.1 Results

Improved Lesk Algortihm GUI Screen shots result

For word- Knife



Give a Sentence

```
A man was beaten and cut with a knife.
```

Ambiguous words

```
man knife beaten cut
```

Word To be disambiguated

```
knife
```

Sense with Lesk Algorithm

```
use a knife on
```

POS tags

```
[['knife' 'n']
 ['man' 'n']
 ['beaten' 'v']
 ['cut' 'v']
```

Sense with Updated Lesk Algorithm

```
a weapon with a handle and blade with a sharp point
```