Combining Lexical and Syntactic Features for

Supervised Word Sense Disambiguation

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Saif Mohammad

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

August 2003

UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of master's thesis by

SAIF MOHAMMAD

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Dr. Ted Pedersen

Name of Faculty Adviser

Signature of Faculty Advisor

Date

GRADUATE SCHOOL

I would like to take this opportunity to thank my advisor Dr. Ted Pedersen for having the belief in me and being generous with his patience and time. I also thank Dr. Rada Mihalcea for her guidance and kind words of encouragement. I am grateful to Dr. Hudson Turner and Dr. Yongcheng Qi for being on my committee and providing valuable feedback. My fellow NLP group members, Bano, Nitin, Sid, Amruta and Bridget, deserve much praise for constantly striving to help me improve and their support through this endeavor.

Contents

List of Figures

List of Tables

Abstract

Most words in any natural language have more than one possible meaning (or sense). Word sense disambiguation is the process of identifying which of these possible senses is intended based on the context in which a word occurs. Humans are cognitively and linguistically adept at this task. For example, given the sentence *Harry cast a bewitching spell*, we immediately understand *spell* to mean *a charm or incantation* and not *to read out letters* or *a period of time*. We can do this via our considerable world knowledge and a fairly limited amount of surrounding context.

However, automatic approaches to sense disambiguation do not have access to our world knowledge, and must take a different approach. The dominant approach at present is to rely on supervised learning, where a human expert provides examples of correctly disambiguated words, and a machine learning algorithm is used to induce a model from these examples. A key issue in such approaches is determining how to represent the context in which the word occurs to the learning algorithm. Pedersen (2001) shows that lexical features (word bigrams in particular) are excellent sources of disambiguation information for a machine learning algorithm. However, there is a large body of previous work in supervised word sense disambiguation that suggest that syntactic features such as part of speech tags and parse structures are also reliable indicators of senses (e.g., McRoy (1992), Ng and Lee (1996)).

This thesis presents a detailed study of the impact of syntactic features in combination with lexical features. We carry out an extensive empirical evaluation using most of the sense-tagged text currently available in the research community. This includes the Senseval-1, Senseval-2, *line, hard, serve* and *interest* data. We find that there is complementary behavior between lexical and syntactic features, and identify several syntactic features that are particularly useful in combination with lexical features. We also introduce a methodology based on comparing the optimal and actual performance of feature sets in order to determine which features are particularly suited to being used in combination, and show that this method leads to improved disambiguation results.

Finally, in the course of part of speech tagging this data, we identified a limitation in the widely used Brill Tagger (1994) that has been corrected via a mechanism known as ”Guaranteed Pre-Tagging” (Mohammad and Pedersen, 2003).

1 INTRODUCTION

Most words in any language have more than one possible meaning. This phenomenon is known as *polysemy* and the different meanings are called *senses* of the word. Word sense disambiguation is the process of identifying the intended sense of a word in written text. The word to be disambiguated is known as the target word. The context consists of the sentence hosting the target word, and perhaps a few surrounding sentences as well. For example, consider the following sentence where *spell* is the target word:

$$\texttt{Harry cast a bewitching } \textbf{spell} \tag{1}$$

spell has many possible senses such as *a charm or incantation, to read out letter by letter* and *a period of time*. Here the context consists of the host sentence. The target word along with its context shall be termed an instance. Thus, the process of word sense disambiguation involves classifying an instance into one of its many senses, based on its context. Although inherent to human cognition, building a computer system adept in word sense disambiguation remains a challenge.

The applications of word sense disambiguation are widespread, Kilgariff [27] describes several, including machine translation, information retrieval and lexicography. Machine translation involves translating a text from one language to another. One of the many reasons why the process is not trivial is polysemy. In order to use a bilingual dictionary which gives translations of a word from one language into another, the intended sense of the word must be determined first. The primary objective of information retrieval is to access the most pertinent information. Consider a query to attain relevant documents. The query terms might individually have many senses. The query result might thus contain documents pertaining to various combinations of the senses of the query terms. However, the relevant documents correspond to a certain combination. For example, if Jack loves the game of cricket and types in the query *cricket bats*, it is likely that apart from documents pertaining to the sport, the query might yield documents pertaining the mammal *bat* and the insect *cricket*. The knowledge of the intended senses of the query terms will thus result in a focussed search, culminating in the most pertinent documents.

There are two broad methodologies to word sense disambiguation: knowledge rich or machine learning. This thesis takes a corpus based learning approach. A knowledge rich approach depends on external sources such as dictionaries and ontologies for the sense discriminating knowledge. Consider the following sentence with *field* as the target word:

$$\texttt{Jack ran across the football } \textbf{field} \tag{2}$$

One way to build a knowledge rich system would be to manually encode substantial amounts of world knowledge and information into the system. Given the sentence above, the system should know that *a piece of land* can be run over but a *sphere of interest* cannot, just to discriminate between the possible senses of *field* . It should be able to infer that the *cricket* related sense corresponding to the concept of *cricket*, cannot be the intended sense, as it is conflicting with the concept of *football* . Based on the information stored within the ontologies, the system must infer that the intended sense of *field* is *an area constructed, equipped, or marked for sports*. Numerous systems have been built for tasks beyond word sense disambiguation in particular domains with laboriously hand coded knowledge bases. Brill and Mooney [11] cite a few, such as the *blocks world* problem described by Winograd [71]. Although very successful in their domains, these systems possess inherent drawbacks. Brill and Mooney [11] point out that creation of these systems entailed extensive domain knowledge. The procedure of building such knowledge bases is expensive, time intensive and error prone since it must be done manually. These systems are brittle, that is, they are designed for a particular task and are not easily scalable to changed or enhanced requirements.

In a machine learning approach the system learns from manually created examples and is then able to classify a new instance with the desired sense. These are empirical approaches where a model gains discriminative knowledge by finding pattern in a large text corpus. They obviate the need for large manually encoded ontologies and are not as brittle. Mitchell [40] defines the concept of learning in his book *Machine Learning* as follows:

> A computer program is said to learn from experience E with respect to some class of tasks T and performance P, if its performance at tasks T, as measured by P, improves with experience E

The task under consideration in this thesis is Word Sense Disambiguation. The performance of the system is evaluated based on the classification of instances for which the intended sense of the target words is known. Experience is gained from a set of instances wherein the target words have been manually tagged with their intended sense in the context. Such a data set is known as sense–tagged data and when used by an automated system to gain experience is known as training data. Thus, the manually encoded information utilized by most supervised word sense disambiguation systems is the sense–tagged data. A learning algorithm is used to induce a model (or classifier) from the training data corresponding to each target word. Occurrences of the target word in new sentences are then classified into appropriate senses by the classifier. It may be noted that since a classifier is learned for every word to be disambiguated, sense-tagged data corresponding to each of the target words is needed for the training. Sense–tagged data can also be used to evaluate the performance of the resulting classifier and is then known as test data. The learning algorithm usually does not utilize the complete instance to learn a classifier. It utilizes that information which is believed to be relevant. For example consider the following sentence with *court* as the target word.

`We settled the` **law suit** `outside the` **court** . (3)

The presence of *suit* immediately after *law* strongly suggests that *court* is used in the judicial sense and not the dating one. The system may thus utilize two word sequences or bigrams in an instance to determine the sense of the target word. The bigrams used are referred to as features. The system may also use unigrams for the purpose, where unigrams are defined as words that occur in the text. Since unigrams and bigrams are part of the text itself, they are known as lexical features.

The learning algorithm acquires better sense discriminating knowledge if relevant features are chosen. The decision tree of bigrams has been shown to perform very successfully by Pedersen [55]. The fact that lexical features, such as bigrams, are easily extracted from data and result in high accuracy, make it a strong baseline to compare results with. However, extensive work by the likes of McRoy [38] and Ng and Lee [53], has shown the utility of syntactic features such as parts of speech and parse structures. Yet, it remains unclear as to how much, if at all, the syntactic features help beyond what is provided by lexical features. Pedersen [56] describes the notion of complementarity and redundancy which is useful to gain insight in this matter. Given two separate sets of features, there will be a certain number of instances which will be correctly tagged by both features sets individually. The ratio of these instances to the total instances quantifies the redundancy in discriminating knowledge provided by the two feature sets. On the other hand, the accuracy obtained by an optimal combination of the two system will be useful in determining the complementarity of the two systems. By optimal combination we mean a hypothetical ensemble technique which correctly identifies the intended sense of an instance, if any one of the feature sets suggests it. This ensemble technique, albeit hypothetical, provides an upper bound to the accuracy achievable by the combination of the two feature sets.

The difference between the optimal combination and the higher of the two accuracies of individual feature sets may used to determine if the combination of the two feature sets is justified.

This thesis uses syntactic features such as parts of speech of words surrounding the target word and the parse tree structure of the sentence housing the target word for word sense disambiguation. We believe that the part of speech features are useful in capturing the local context of the target word which parse features help identify features which involve words that are further away from the target word but syntactically related. We study the contribution of syntactic features to word sense disambiguation when combined with Lexical features. We performed an extensive set of experiments on the Senseval-2, Senseval-1, *line, hard, serve* and *interest* data which together comprise of almost all the sense tagged text available in the research community. We found that there is a significant amount of complementarity across lexical and syntactic features. We identify numerous syntactic features which are particularly useful in sense disambiguation. Apart from overall results for Senseval-1 and Senseval-2 data we provide a break up of the performance of the various features for each part of speech. We specifically point out features which are useful disambiguate words belonging to particular parts of speech.

In the process of running these experiments, we have part of speech tagged and parsed the Senseval-2, Senseval-1, *line, hard, serve* and *interest* data using the `posSenseval` [48] and `parseSenseval` [47] packages which utilize the Brill Tagger and Collins Parser, respectively. In the course of the tagging we identified a limitation of the Brill Tagger and overcame it with a mechanism known as *Guaranteed Pre-Tagging* [42], which is also discussed in this thesis.

2 BACKGROUND

2.1 Decision Trees

2.1.1 Decision Trees as Classifiers

A decision tree is a kind of classifier. Given an instance, it assigns a class to it by asking a series of questions about it. Except for the first, the question asked depends on the answer to the previous question. For example, consider the task of choosing toys for four year old Max. A decision tree may be used to automate the process. Given a new toy, the decision tree may assign the classes *Love It, Hate It* or *So So* to it, thereby predicting if Max would love the toy, hate it or not really have an opinion on it. Decision trees have an inverted tree structure, with interconnected nodes as shown in figure 1. Every node is associated with an attribute or question about the instance being classified. In figure 1, nodes are marked by rectangles, while the circles represent leaves. Each leaf is associated with a class. The various nodes and leaves are connected to each other by branches. The branches have been shown as directed straight lines.

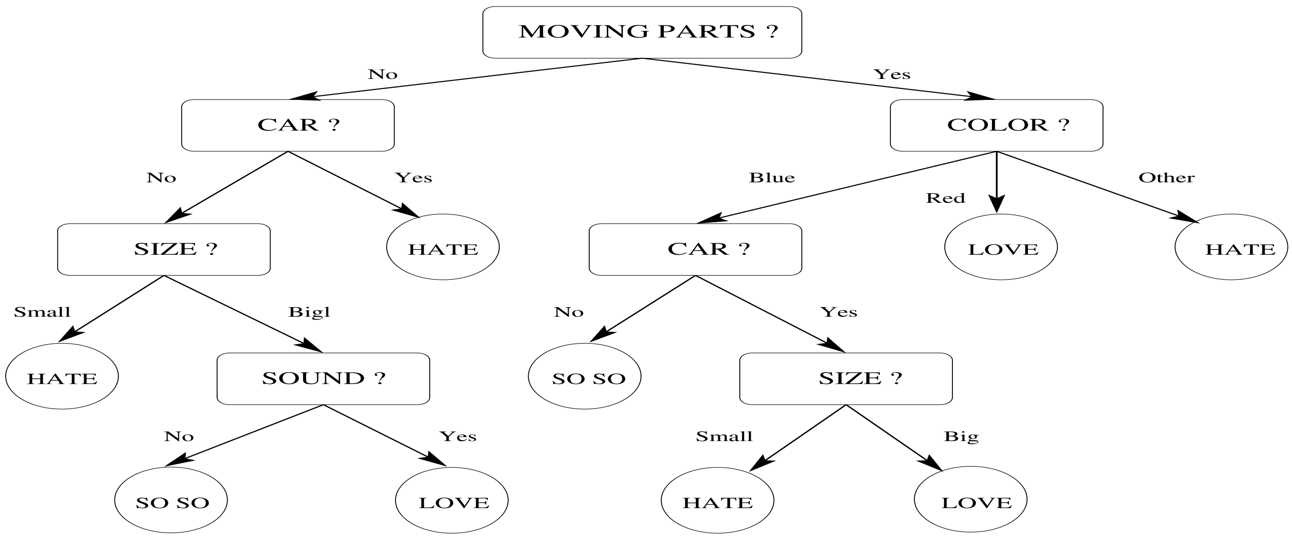

Figure 1: Decision Tree to Choose Toys

Each possible value of the attribute is associated with a branch emanating from that node. For example, figure 1 has a node with the associated attribute *COLOR*. It has three branches emanating from it which are associated with the values *Blue, Red* and *Other*. At any given node n, the instance being classified takes a branch to arrive at the next node or leaf. The branch corresponding to the value of the attribute associated with *n* is chosen. For example, if COLOR is the current node and the toy being classified is *Blue*, the corresponding branch is chosen to arrive at the node *CAR*. Then the question '*Is the toy a car?*' is asked. If yes, the branch corresponding to 'yes' is taken to arrive at the node *size*. Depending on the *size* of the toy appropriate branch is taken to arrive at the leaf. If the toy is big, we arrive at the leaf node *LOVE* and the tree predicts that Max will love the toy. Given a new instance, the afore mentioned process starts at the root and is repeated until we reach a leaf. The class associated with the leaf is then assigned to the instance. Figure 2 shows how toys X and Y are classified by the decision tree depicted in figure 1. The attribute values of toys X and Y are listed in table 1.

Table 1: Attributes values of instances X and Y

Instance	MOVING PARTS	COLOR	CAR	SIZE	SOUND
X	*Yes*	*Blue*	*Yes*	*Big*	*No*
Y	*No*	*Other*	*No*	*Big*	*No*

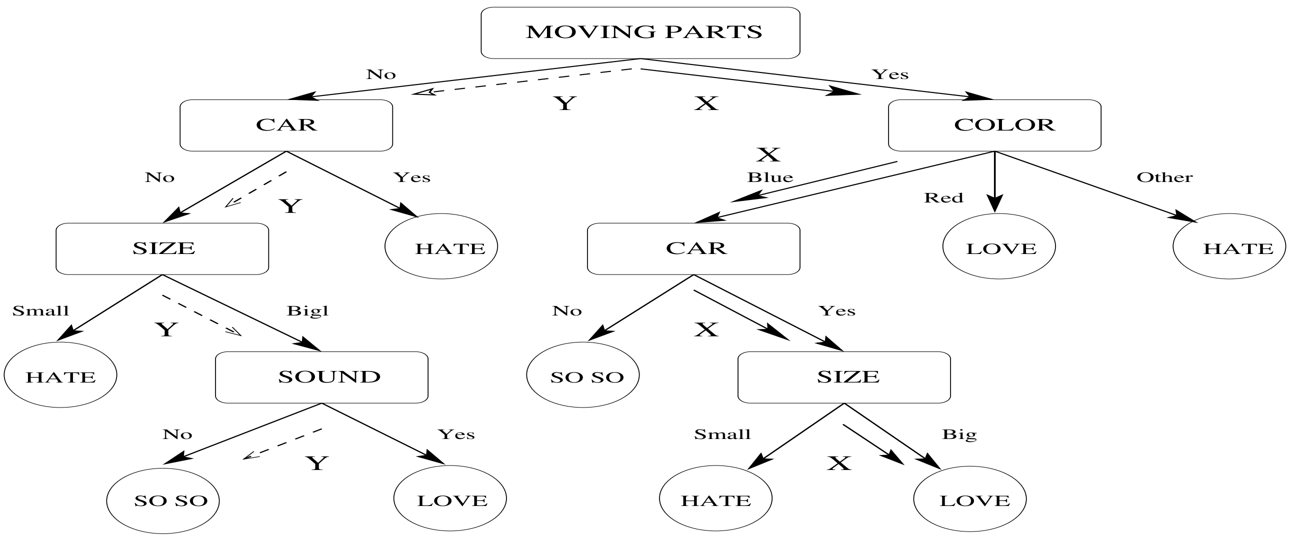

Figure 2: Classification of Toys X and Y

Consider how toy X is classified by the decision tree. We start from the root node which is *MOVING PARTS*.

We ask the question *Does X have moving parts?* Based on the answer, *Yes*, we choose the appropriate branch to arrive at node *COLOR*. We then ask *What is the color of X?* which leads us to node *SIZE*. We then ask *What is the size of X?*. Since X is big, we choose the corresponding branch and arrive at a leaf. The iterative process is stopped as we have reached a leaf and the class associated with the leaf, *LOVE* in this case, is assigned to the instance. The decision tree thus predicts that Max will like the big blue car X. Toy Y is classified in a similar manner. Due to the nature of the decision tree structure, it is useful only with features which have discrete values. Continuous valued features if associated with a node will lead to the node having infinite branches emanating from it. Continuous features might still be incorporated if made discrete. Due to the feature oriented structure of the decision tree, it is ideally suited to classification problems where the instances can be easily and adequately described by a finite number of discrete features.

2.1.2 Learning Decision Trees

A decision tree is learned from a set of instances for which the classification is known, using a suitable learning algorithm. This set of examples is known as the training set. Returning to the toy example, if the attributes of the toys which Max has liked and disliked in the past is known, a decision tree can be learned. Two of the most successful decision tree learning algorithms are C4.5 [63] and CART [7], both of which are based on ID3 [61].

Given a set of training instances, the ID3 algorithm learns a decision tree that classifies the training examples as correctly as possible. The learning of the tree is a top down and greedy approach. The attribute associated with the root node is chosen first, based on the complete set of training examples. The root node is the first question to be asked about an instance. The attributes of the lower nodes are learned progressively. Thus, the tree is learned from root to leaves.

Each node is chosen based on a set of training instances and candidate attributes. The root node is chosen based on all the available training instances and its set of candidate attributes encompasses all the attributes. The attribute (one of the candidate attributes) which best classifies the associated instances is chosen to be the node. Branches are created going downward from this node to its child nodes. Each branch corresponds to a distinct value of the chosen attribute. Instances associated with the parent, are passed down along these branches as long as they have the same attribute value corresponding to the branch. Child nodes are created that terminate these branches, based on the subset of the training instances associated with the parent, which are passed down the branch. All candidate attributes associated with the parent, minus the chosen attribute for the parent, form the candidate attributes out of which the child node is chosen. The process of learning as described above is repeated recursively for each of the child nodes leading to simultaneous generation of nodes in various paths starting from the root node. It may be noted that there is a recursive partitioning of the training data along every path, entailing the requirement of a large training data for effective learning. If all the instances associated with a node belong to the same class, *say X*, a leaf node is created and no further child nodes are generated along that path. Class *X* is assigned to the leaf. Thus a particular path from root to leaf need not have all the attributes as nodes. Further, there may be attributes which do not figure in the learned decision tree at all. If all the attributes have been used up as nodes along a path, the class, *say m* , most common amongst the examples associated with the node is determined. A leaf node assigned the class *m* is created and no further nodes are generated along that path.

The ID3 learning is greedy in the sense that the attribute which is most effective in classifying the training instances available is chosen as the node. The effectiveness of an attribute in classifying a given set of examples is quantified using the concepts of Entropy and Information Gain. The entropy of a set of instances gives an idea of the distribution of instances as per the various possible associated classes. If all instances belong to the same class, there is said to be a 0 entropy. If instances are equally distributed among all the possible classes, the entropy is 1 with minimal uncertainty. A low entropy value indicates that the attribute is able to accurately distinguish between the classes and is thus desirable. Entropy is calculated using the following formula:

$$Entropy(S) = \sum_{i=1}^{c} -p_i \ * \ log_2 \ (p_i) \tag{4}$$

c is the number of possible classifications and p_i is the fraction of instances of *S* which are of class *i*. Information gain is defined as the difference in entropy of the parent node and the sum of entropies of all its child nodes. The entropy of the child nodes is normalized based on the number of examples associated with the child node. Mathematically, the information gain *Gain(S,A)* for an attribute *A* of a parent node associated with the set of examples *S* is represented as follows:

$$Gain(S, A) = Entropy(S) - \sum_{v \epsilon Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{5}$$

where *v* takes on the different values of *A* associated with the various branches emanating from the parent node. S_v stands for the subset of instances of *S*, which have value *v* for attribute *A*. The attribute with the highest information gain is chosen to represent the node.

2.1.3 Overcoming the Drawbacks of ID3

While it represents the start of modern decision tree learning, the ID3 algorithm has a number of limitations. It can be used only if the features have discrete values. The values for all the attributes of the instances must be known. ID3 does not account for instances with missing values or continuous valued attributes. Yet another potential drawback of the ID3 algorithm is over-fitting. The classifier learned might be too specific for the training data, that is, apart from capturing the general features of instances in each class, the classifier could have captured idiosyncrasies specific to the training set which are not really representative of the problem in general. Thus, the decision tree memorizes the examples in the training set, thereby classifying them well but fails to do nearly as well for new test data. The C4.5 [63] algorithm was intended to overcome these limitations.

Consider a node for which we need to choose a feature based on Information Gain, such that one of the associated instances has an unknown value for a feature F. Let F be a boolean feature which can take one of two values 0 an 1. The C4.5 [63] algorithm determines the probabilities of F having each of the possible values. These probabilities are calculated based on simple counts of associated instances (N) for which the value of F is known. The probability assigned to a particular value, say 0, is equal to the ratio of instances in N which have F with value 0 upon the total number of instances N. Thus pruning removes certain sub-trees whose absence does not reduce the accuracy of the classifier significantly. There are many pruning techniques such as the reduced error pruning [62] and rule post pruning [63], however, it is the latter which is used in C4.5. Since, all our experiments deal with discrete features, we shall not delve into details of how continuous valued features are handled by C4.5. In this thesis we use the C4.5 algorithm as implemented by Waikato Environment for Knowledge Analysis *Weka*[23][72], to learn decision trees for each word to be disambiguated. The Weka implementation is in Java and is based on a modified version of the C4.5 algorithm known as C4.5 Revision8.

In general decision tree learning is sensitive to the training examples. Slight variations in the training set may create significantly different trees. A few spurious instances might mislead the learning process creating an erroneous tree structure. This may be overcome by bagging [6], which is a technique of generating multiple classifiers by random sampling of the training data set with replacement. That is, multiple classifiers are learnt based on different sets of training data or samples. The individual sets of training data are created from the total available training instances as follows. If the training data has *N* instances, then *N* draws are made on it. Each draw selects one instance to be part of a sample. A copy of the instance is put back in before the next draw. Thus, after *N* draws, *N* instances are chosen to be part of a sample. It may be noted that there might be multiple copies of some instances in a sample. Certain instances from the training data may not be selected to be part of the sample at all, which is intended in order to smooth out spurious or misleading examples. Multiple samples are created this way, each acting as training data for a classifier. One vote is assigned to the classification of an instance by each classifier. The class which gets the maximum votes is chosen. Quinlan [64] has shown that the accuracy of decision trees can be improved by bagging [6].

2.1.4 A Word Sense Disambiguation Tree

Word sense disambiguation involves classifying an occurrence of a word in its context, into one of its many discrete senses. Each such instance, composed of the word and its context, can be represented by a rich set of discrete features which may be effectively utilized by decision trees. Pedersen [55] shows that varying the basic learning algorithm yields little variations in the achieved accuracies for a given task of word sense disambiguation. He states that identifying the most useful features to use for disambiguation is of greater significance. He believes that decision tree learning for word sense disambiguation is reasonably accurate and has further advantage of showing relationship amongst features via the structure of the decision tree.

Since each of the words has multiple possible senses, the task of word sense disambiguation is a multi–class problem. All features used in the thesis are discrete and boolean which take on the values 0 or 1. Features which were originally multi-valued have been split into multiple boolean features as follows. Consider a feature F which has three possible values - a, b and c. It is split into three features - F1, which determines if ‘a’ occurs or not, F2 which determines if ‘b’ occurs or not and F3, which determines if ‘c’ occurs or not. Also, as mentioned earlier, given a new instance, the C4.5 algorithm calculates a probability for each of the intended senses. The sense which has the highest probability is chosen as the intended sense. However, if the difference in the top two probabilities is less than 0.05, the instance is tagged with both senses. If only one of the two is listed as the correct sense of the system, this particular assignment of senses gets a score of 0.5. Figure 3 illustrates a generic decision tree for word sense disambiguation. Choosing the right features and efficiently capturing these features from a given data set plays a vital role in the success of decision trees. The following sections detail some of the potentially useful features for word sense disambiguation.

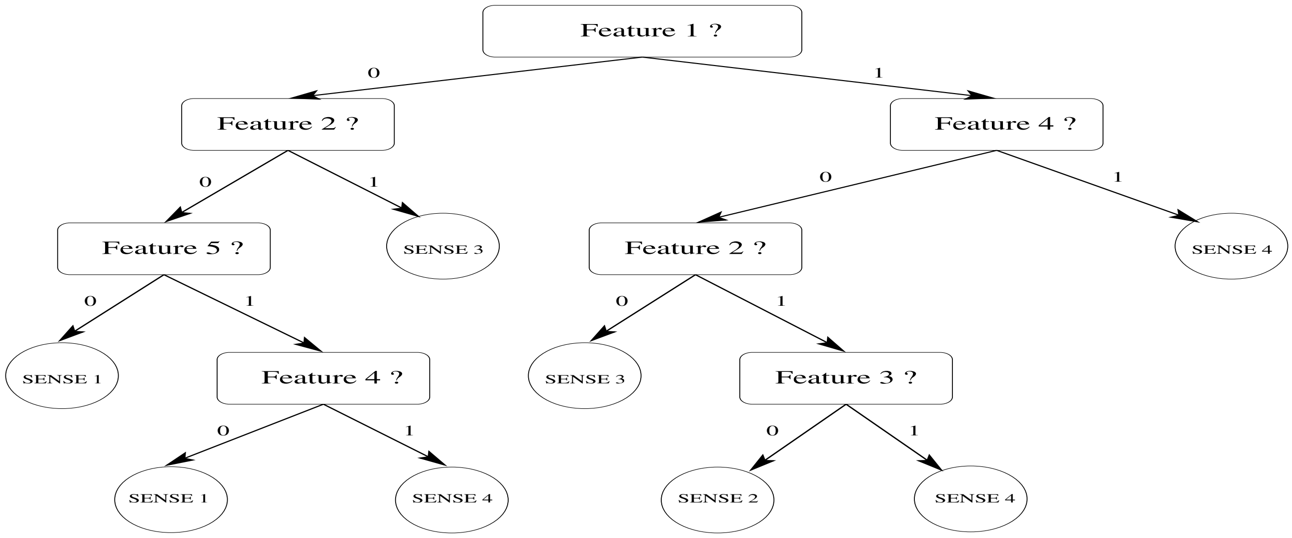

Figure 3: A Generic Decision Tree for Word Sense Disambiguation

2.2 Lexical Features

In strictly corpus based, supervised word sense disambiguation, the classification system relies solely on the knowledge gained from training instances to assign senses to the words being disambiguated or target words in the test data. The features or attributes of an instance which help identify the intended sense of target word are identified. Both training and test instances are represented by these features and corresponding values for the instance. Numerous features may be used, based on different knowledge sources such as collocations, part of speech and parse structures. Features which may be captured directly from the text are known as lexical features and are described in the following sub-sections.

2.2.1 Surface Form

A word we observe in text is known as surface form. It may be broken down into a stem and a possible prefix, infix and suffix. The stem, which is also known as the root word, is the base form of the word which has meaning and cannot be broken up into multiple tokens. Suffixes, prefixes and infixes together form a general class known as affixes. They are a set of characters which do not have meaning on their own but add or change the meaning of a stem when attached to it. Suffixes, prefixes and infixes may attach at different positions to the stem to form the different surface forms associated with the stem. A word without any affixes is also considered to be a distinct surface form of the stem. Consider the verb *qualify*. It has no affixes, and hence both, the surface form and stem, are *qualify*. It has the following meanings in Merriam Webster's Online Dictionary:

A-I: to reduce from a general to a particular or restricted form : MODIFY

A-II: to make less harsh or strict : MODERATE

A-III: to characterize by naming an attribute : DESCRIBE

A-IV: to fit by training, skill, or ability for a special purpose

A-V: to declare competent or adequate : CERTIFY

A-VI: to be or become fit (as for an office) : meet the required standard

A-VII: to acquire legal or competent power or capacity

A-VIII: to exhibit a required degree of ability in a preliminary contest

A prefix is a set of characters attached to the start of a stem to modify or change its meaning. For example the prefix *pre* when attached to *qualify* forms *pre-qualify*, which is a different surface form of the stem *qualify*. It adds one of the following meanings to the sense of *qualify*:

B-I: earlier than : prior to : before

B-II: preparatory or prerequisite to

B-III: in front of : anterior to

A suffix is a set of characters attached to the end of a word to modify or change its meaning. For example the suffix *ied* when attached to *qualify* forms *qualified* which has the meanings listed below. *qualified* is yet another surface form of the stem *qualify*.

C-I: fitted (as by training or experience) for a given purpose : COMPETENT

C-II: having complied with the specific requirements or precedent conditions (as for an office or employment) : ELIGIBLE

C-III: limited or modified in some way

Other surface forms of *qualify* are *qualification, qualifiable* and *post-qualify*. It may be noted that each surface form has at least a slightly different meaning from the rest, however, the granularity of sense distinctions made in most dictionaries is broader. Thus a particular sense in a dictionary may encompass multiple surface forms of a word. For example, sense C-I of *qualified* is very close in meaning to sense A-IV of *qualify* and a dictionary might have an entry for just one sense corresponding to both. Conversely, a surface form may have one or more senses and not necessary all the senses corresponding to all the surface forms of the stem. Thus, the knowledge of the exact surface form in an instance will aid in restricting the possible senses of the word. For example consider a dictionary with the following coarser senses for *qualify* such that sense C-I and C-II of *qualified* come under sense D-III and sense C-III comes under D-II.

D-I: modify or moderate

D-II: describe

D-III: make fit or eligible

Then given that the surface form is *qualified* we are certain that it has sense D-I or D-III and not sense D-II corresponding to *describe*.

Additionally, given a sense, some surface forms occur more frequently than others. This information may be utilized to identify the sense which is most likely given a morphological form of a word and which sense is rarely associated with that form. For example, the training data used in the Senseval-2 exercise has instances corresponding to the verb *treat* (Senseval-2, held in the summer of 2001, was an event where numerous word sense disambiguation systems from across the world were evaluated on a common data set). The surface form *treating* occurs in 15 of the instances, 10 of which are tagged with the sense *treat%2:29:00::*. Three of the instances are tagged with the sense *treat%2:31::*, while one each to *treat%2:30::* and *treat%2:41::*. Given a test instance with the target word having the surface form *treating*, the sense *treat%2:32:00::* may be ignored at the outset. Further, if the other features fail to confidently identify the intended sense, the surface form frequency information may be used to tag the instance with the sense *treat%2:29:00::*, which was far more frequent in the training data for the given surface form, than the rest. The surface form of a word can be easily captured from the given instance. Information about which sense is common and which is rare for a given surface form, may be learnt from a training set or acquired directly from a dictionary.

2.2.2 Most Frequent Sense

The number of times a word is used in its various senses is not uniform. Words tend to be used more commonly in some senses than others. For example the word *dam* in its noun form has two senses listed in Table 2. The sense pertaining to *the body of water* is much more common than the *animal* sense in nearly all domains of text. This phenomenon is reflected in bodies of text, as well. The distribution of the instances in Senseval-2 test data, which have the adjective *blind* as the target word is shown in Table 3. If all other features fail to confidently identify the intended sense, choosing the most frequent sense is expected to give an accuracy better than that obtained by a random choice. The most frequent sense of a word can be identified by a simple count of the instances with the target word in various senses. From the details in table 3, it is evident that given a random test instance with the target word *blind*, sense *blind%3:00:00::* would be a good guess of the intended sense.

Table 2: Senses of *dam*

Sense	**Usage**
Barrier to a body of water	common
Female parent of a quadruped	very rare

2.2.3 Unigrams

Unigrams are individual words that occur in the context of an instance. Consider the following sentence:

$$\texttt{the judge dismissed the case} \tag{6}$$

It has the unigrams *the, judge, dismissed* and *case*. It may be noted that the unigrams *judge* and *dismissed* suggest that *case* has been used in the *judicial* sense and not the *container* sense. Unigrams like *the, it, on, at* and *of* which are found in almost all sentences and which occur independent of the intended sense of words

Table 3: Instance Distribution of *blind*

Sense	Frequency	Percentage
blind%3:00:00::	47	85.5%
blind%5:00:00:irrational:00	6	10.9%
blind%5:00:00:unperceptive:00	1	01.8%
blind_man%1:18:00::	1	01.8%
TOTAL	**55**	**100%**

are included in a list of words that are thought to be unimportant for disambiguation. This list is known as the stop list and all unigrams which occur in the stop list are ignored. Those unigrams which occur at least a few times (U_{min} say) in the context and which are not listed in a pre-determined stop list, are considered as features. Note that the surface form feature which considers only the target word is a subset of the unigrams feature. Unigrams consider not just the target word but other words in the context as well.

2.2.4 Bigrams

Bigrams are two-word sequences in the context of an instance. Like Unigrams, the words forming the Bigram may or may not include the target word. Consider the sentence:

`the interest rate is lower in state banks` (7)

It is made up of the bigrams *the interest, interest rate, rate is, is lower, lower in, in state* and *state banks*. We notice that the bigram *interest rate* suggests that *bank* has been used in the *financial institution* sense and not in the *river bank* sense. Like unigrams, an appropriate stop list may be created to ignore bigrams which occur commonly in text and independent of the intended sense of words. Only those bigrams may be considered which occur at least a certain pre-ascertained times (B_{min} say), in the training data. Bigrams composed of words which may not occur consecutively and have one or more words between them may also be considered by allowing window size of $\pm$n words in between. This helps capture bigrams like *rate interest* which would otherwise be lost due to the presence of *of* in the phrase *rate of interest*. Word pairs which have more than *n* words between them are not considered.

2.2.5 Collocations

Words in an ordered sequence, which tend to occur together more often than random chance, form a collocation. For example *car chase* or *bread and butter*. The former is a compositional collocation as its meaning is understood by the conjunction of the meanings of its constituent words, *car* and *crash*. The latter is a non-compositional (idiomatic) collocation which takes on a meaning quite different from its constituents. Collocations are strong indicators of the sense of words constituting it. For example, given any instance with the collocation *bread and butter*, *butter* will always be used in the *food* sense and not the *flattery* sense. This demonstrates Yarowsky's *One Sense per Collocation* hypothesis [73]. Additionally, collocations are also strong indicators of the topic of discussion and hence may be useful in suggesting the intended sense of words which may not be part of the collocation but are in the same instance. For example, the collocation *interest rate* suggests that the topic of discussion is related to money. If the word *bank* is used in the same instance then it is likely that it has been used in the *financial institution* sense and not the river bank sense. The inventory of collocations is either accessed directly from a dictionary or extracted from a text corpus. Those word sequences are chosen as collocations which occur more often in the text than would be expected by random chance.

2.2.6 Co-Occurrences

Co-occurrences are pairs of words that tend to occur in the same context, not necessarily in any order and with a variable number of intermediary words. For example *testify* and *court*. Like collocations they too are indicative of a sense of the constituent words. *court* for example could be *the place of justice* or *solicit for marriage*. But the presence of *testify* in the context suggests that it is being used to refer to *the place of justice*. A list of co-occurrences having the target word as one of the constituents can be taken from a dictionary or extracted from a sense tagged text. A body of text where certain words are marked with their intended sense by a human is known as a sense tagged text. extracting collocations from sense tagged text involves the application of conditional probability as defined by the formula below and entails the following stipulations:

$$\mathrm{Cp(i,k)} = \mathrm{N}_{i,k} \,/\, \mathrm{N}_k \qquad \text{(Ng and Lee [53])}$$

Cp(i,k) is the conditional probability of the target word being used in sense-i, given that the word K co-occurs with it. $N_{i,k}$ is the number of times the word k has co-occurred with the target word and the sense of the target word has been i. And N_k is the number of times the word k has co-occurred with the target word. Following stipulations are usually applied: Only those pairs with a conditional probability greater than a pre-ascertained minimum ($Cp(i,k)_{min}$) are chosen. The pair must co-occur in a certain minimum(N_{kmin}) number of sentences. An upper limit(Max) to the number of co-occurrences per word is set; the most suggestive chosen.

Divergent to the case of collocations though, co-occurrences invariably lack non-content words. This is apparent from the formula, as non-content words like *a, in and the* occur in all sentences and hence occur with the target word, in most of its senses. Note, content words can loosely be defined as all words other than articles, prepositions and determiners.

2.3 Syntactic Features

2.3.1 Part of Speech Features

Words may be classified into different semantic/syntactic classes such as nouns, verbs and adjectives, known as the parts of speech. Assigning the appropriate part of speech to the words in a text is known as part of speech tagging. Consider the following sentence.

`Jack will chair the meeting` (8)

It may be part of speech tagged as shown below, given, NNP stands for proper noun, MD for modal, VB for verb, DT for determiner and NN for noun.

`Jack/NNP will/MD chair/VB the/DT meeting/NN` (9)

Instances of words in different parts of speech have disparate possible senses. Consider the the word *float*. It can be a noun or a verb depending on the context. The Senseval-1 training data has the noun *float* tagged with 8 senses and the verb *float* with 15 (Senseval-1 exercise, held in the summer of 1998 was a word sense disambiguation event preceding Senseval-2). Given a test instance with *float* as the target word, if we know that *float* was used as a noun, we straightaway eliminate the 15 verb senses from consideration. More generally, if we know the part of speech of the target word, we restrict the possible senses of the word to the senses corresponding to that part of speech. Most word sense disambiguation systems assume resolution of part of speech of the target word, at least up to the broad part of speech level. That is, the target words will be marked to be a nouns, verbs adjectives and so on before fed as input to the system. The system then learns a classifier for the word – part of speech combination. For example a classifier is learnt for the noun *float*. Yet another classifier is learnt for the verb *float*. Thus, part of speech of the target word is inherently utilized by these systems.

A word used in different senses in different sentences is likely to have words with different parts of speech around it. Further, the two or three part of speech tag sequences around the target word are also likely to be different. Consider the following sentences, given that the target word is *turn*. The part of speech of each token follows the forward slash attached to it.

$$\texttt{Why/WRB did/VBD Jack/NNP } \mathbf{turn}/\texttt{VB } \mathbf{against/IN\ his/PRP\$}\ \texttt{team/NN ?/.} \tag{10}$$

$$\texttt{Why/WRB did/VBD Jack/NNP } \mathbf{turn}/\texttt{VB } \mathbf{left/VBN\ at/IN\ the/DT}\ \texttt{crossing/NN ?/.} \tag{11}$$

Notice that the set of part of speech tags to the right of the target word is significantly different in the two instances, which have different intended meanings of *turn*. The first sentence has the sense corresponding to *changing sides/parties* while the second has the sense of *changing course/direction*. Thus the part of speech tags of the word to the right of the target word - P_1, and the part of speech of the word two positions to the right of the target word - P_2, may be used as features to aid word sense disambiguation. The same may be true for the part of speech tags to the left of the target word in certain other contexts. Thus, P_{-1}, P_{-2} and so on may be used as features as well. In this notation, the negative indicates words to the left of the target word. The exact sequence of these tags is also likely to differ when the word is used in different senses and so the sequence information may be used as a feature as well. In the sentences above, the sequence of part of speech tags P_1 and P_2 helps identify the intended sense. We shall refer to this sequence as P_1P_2, a mere concatenation of the symbols identifying the individual parts of speech. Sequences such as $P_{-2}P_{-1}$, $P_{-1}P_0P_1$ and $P_0P_1P_2$ are some more examples of the possibly useful features. Note, P_0 stands for the part of speech of the target word. As we go away from the target word, on either side, the smaller is the probability of the part of speech tags being affected by the usage of the target word. It is generally believed that the range of influence of the target word on parts of speech of its neighbors stretches up to one or two and possible three, positions away from the target word.

To summarize, words in different senses are likely to be surrounded by disparate part of speech tag sets.

This information may be captured by part of speech features and utilized to classify new instances of the word.

2.3.2 Parse Features

Identifying the various syntactic relations amongst the words and phrases within a sentence is known as parsing. A phrase is a sequence of words which together has some meaning but is not capable of getting across an idea or thought completely. For example, *the deep ocean*. The word within the phrase which is central in determining the relation of the phrase with other phrases of the sentence is known as the head word or simple head of the phrase. The part of speech of the head word determines the syntactic identity of the phrase. The aforementioned phrase has *Ocean* as the head, which is a noun. The phrase is thus termed a noun phrase. A parser is used to automatically parse a sentence and identify the constituent phrases and the head words of these phrases. Consider the sentence:

$$\texttt{Harry Potter cast a bewitching spell} \tag{12}$$

Some of the aspects a parser might identify are that the sentence is composed of a noun phrase *Harry Potter* and a verb phrase *cast a bewitching spell*. The verb phrase is in turn made of a verb *cast* and a noun phrase *a bewitching spell*. We shall call the verb phrase parent (phrase) of the verb *cast* and the noun phrase *a bewitching spell*. Conversely, the verb and the noun phrase shall be referred to as child (phrases) of the verb phrase. The parsed output will also contain the head words of the various phrases and a hierarchical relation amongst the phrases depicted by a parse tree. Figure 4 depicts a sample parse tree for the above sentence.

Following is a description of the various syntactic features based on the parsed output of a sentence that are used for word sense disambiguation in this thesis.

2.3.3 Head Word of the Phrase

The head word of a phrase is suggestive of the broad topic of discussion. It is expected that the words in the phrase have senses pertaining to the same topic. Consider the instance from Senseval-2 training data shown in Figure 6 (only the sentence housing the target word is shown for brevity, the other sentences of the context being ignored). Figure 7 shows the parsed output of the sentence in Figure 6.

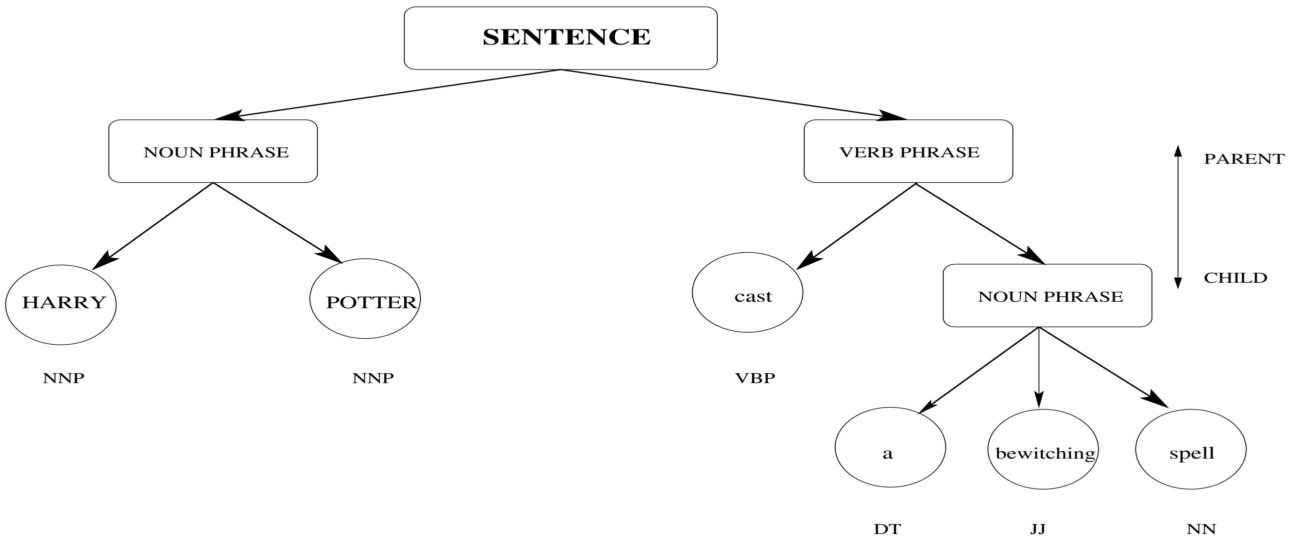

Figure 4: Sample Parse Tree

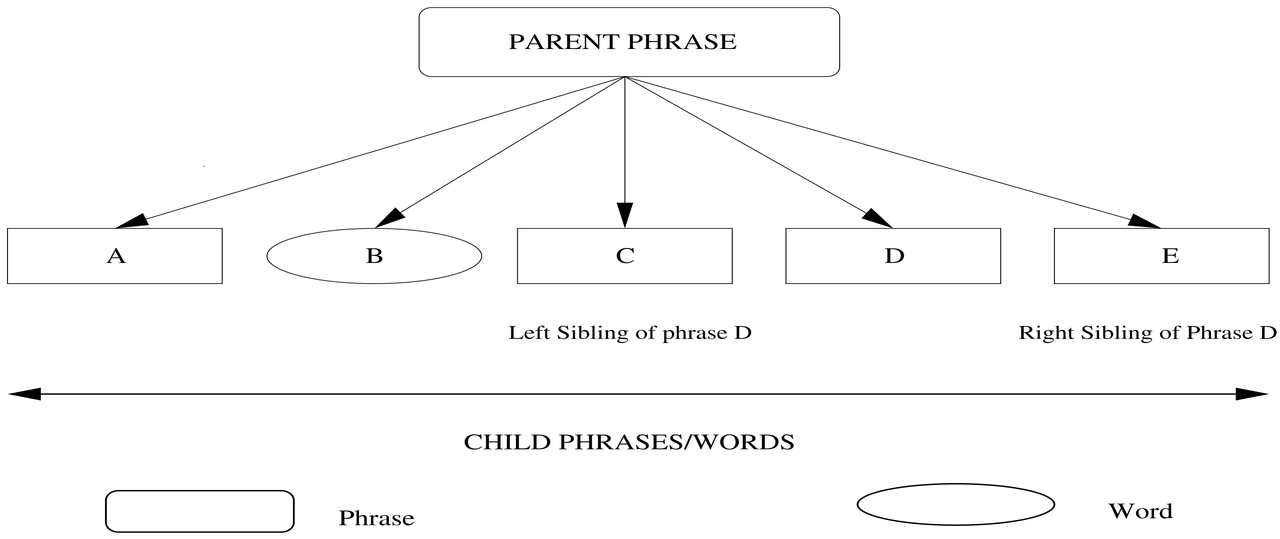

Figure 5: Sample Parse Tree

```
<instance id=”art.40078” docsrc=”bnc_A6E_1305”>
<answer instance=”art.40078” senseid=”art_school%1:06:00::”/>
<context>
I was proud of getting accepted, particularly at St Martin’s because it was such a good
<head>art</head> school, but Fine Art seemed a fast road to no where
</context>
</instance>
```

Figure 6: Head Word of a Phrase: Instance from Senseval-2 Training Data.

```
<instance id=”art.40078” docsrc=”bnc_A6E_1305”>
<answer instance=”art.40078” senseid=”art_school%1:06:00::”/>
<context>
<P=”TOP˜was˜1˜1”> <P=”S˜was˜3˜1”> <P=”S˜was˜2˜2”> <P=”NPB˜I˜1˜1”> i <p=”PRP”/>
</P> <P=”VP˜was˜2˜1”> was <p=”VBD”/> <P=”ADJP˜proud˜2˜1”> proud <p=”JJ”/>
<P=”PP˜of˜2˜1”> of <p=”IN”/> <P=”SG-A˜getting˜1˜1”> <P=”VP˜getting˜3˜1”> gett ing
<p=”VBG”/> <P=”VP-A˜accepted˜1˜1”> accepted <p=”VBN”/> , <p=”PUNC,”/> </P>
<P=”SBAR˜because˜3˜2”> <P=”ADVP˜particularly˜2˜1”> particularly <p=”RB”/> <P=”PP˜at˜2˜1”>
at <p=”IN”/> <P=”NPB˜’s˜3˜3”> st <p=”NNP”/> martin <p=”NNP”/> ’ s <p=”POS”/> </P>
</P> </P> because <p=”IN”/> <P=”S-A˜was˜2˜2”> <P=”NPB˜it˜1˜1”> it <p=”PRP”/>
</P> <P=”VP˜was˜2˜1”> was <p=”VBD”/> <P=”NPB˜school˜5˜5”> such <p=”PDT”/> a
<p=”DT”/> good <p=”JJ”/> <head>art</head> <p=”NN”/> school <p=”NN”/> , <p=”PUNC,”/>
</P> </P> </P> </P> </P> </P> </P> </P> </P> </P> but <p=”CC”/>
<P=”S˜seemed˜2˜2”> <P=”NPB˜Art˜2˜2”> fine <p=”NNP”/> art <p=”NNP”/> </P>
<P=”VP˜seemed˜2˜1”> seemed <p=”VBD”/> <P=”NP-A˜road˜3˜1”> <P=”NPB˜road˜3˜3”> a
<p=”DT”/> fast <p=”JJ”/> road <p=”NN”/> </P> <P=”PP˜to˜2˜1”> to <p=”TO”/>
<P=”NP-A˜nowhere˜2˜1”> <P=”NPB˜nowhere˜1˜1”> nowhere <p=”RB”/> </P>
</context>
</instance>
```

Figure 7: Head Word of a Phrase: Parsed structure of the sentence from Figure 6.

We notice that the target word *art* is part of a noun phrase *such a good art school ,*. The head word of this phrase is *school* which is clearly related to the *art_school%1:06:00::* sense of art. As is evident, the head words of a phrase may help restrict the possible senses of word (if not uniquely identify it). This argument is supported by the *One Sense per Discourse* hypothesis proposed by Gale, Church and Yarowsky [22]. It may be noted, given that the target word is an adjective, the head of the phrase housing the target word is expected to be the noun which is qualified by the adjective. Besides the head word, the part of speech of the phrase encompassing the target word, is also used as a feature. This is to examine if the part of speech of the phrase housing the target word is useful in partitioning the possible senses of the target word.

2.3.4 Head Word of the Parent Phrase

Similar to the head word of a phrase, the head word of parent phrase of the phrase which houses the target word is suggestive of the broad topic of discussion. It is expected that the words in the child phrases have senses pertaining to the same topic. Consider the instance from Senseval-2 training data shown in Figure 8 (only the sentence housing the target word is shown for brevity, the other sentences of the context being ignored). Figure 9 shows the parsed output of the sentence in Figure 8.

<instance id="channel.40187" docsrc="bnc_CRM_4307">
<answer instance="channel.40187" senseid="channel%1:10:00::"/>
<context>
On the basis of these studies, Ca 2+ is suggested to play a central role in photorecovery
and light adaptation, not only by regulating guanylate cyclase, possibly through recoverin,
but also by modulating the cGMP-gated <head>channel</head> through calmodulin
interaction with the 240K protein.
</context>
</instance>

Figure 8: Head Word of the Parent Phrase: Instance from Senseval-2 Training Data

Notice that the target word *channel* is part of a noun phrase *the cgmp-gated channel*. The head word of this phrase is *channel* which on its own is not of much use in discriminating amongst the various senses

<instance id=”channel.40187” docsrc=”bnc_CRM_4307”>
<answer instance=”channel.40187” senseid=”channel%1:10:00::”/>
<context>
. . .
. . .
<P=”CONJP˜but˜2˜1”> but <p=”CC”/> also <p=”RB”/> </P> <P=”PP˜b y˜2˜1”> by
<p=”IN”/> <P=”SG-A˜modulating˜1˜1”> <**P=”VP˜modulating˜4˜1”**> **modulating**
<p=”VBG”/> <**P=”NPB˜channel˜3˜3”**> the <p=”DT”/> cgmp-gated <p=”JJ”/>
<**head**> **channel** <**/head**> <p=”NN”/> </P> <P=”PP˜through˜2˜1”> through <p=”IN”/>
<P=”NPB˜interaction˜2˜2”> calmodulin <p=”NN”/ > interaction <p=”NN”/> </P> </P>
<P=”PP˜with˜2˜1”> with <p=”IN”/> <P=”NPB˜protein˜3˜3”> the <p=”DT”/> 240k
<p=”CD”/> protein <p=”NN”/> . <p=”PUNC.”/> </P> </P> </P> </P> </P>
</context>
</instance>

Figure 9: Head Word of Parent Phrase: Parsed structure of sample sentence from Figure 8.

of the target word. The parent of this noun phrase is a verb phrase which has the following constituents: a verb *modulating*, the noun phrase which holds the target word and two prepositional phrases. The head word of this verb phrase is the verb *modulating* which suggests the *transmission channel* sense of the target word *channel* which corresponds to the sense ID *channel%1:10:00::*. Note that in this case, the verb object relationship is captured by the parent phrase head word and target word. The noun *channel* is the object of the verb *modulating*. Thus, the head word of a parent phrase of the target word may help restrict the possible senses of word if not uniquely identify it. This argument is also supported by the *One Sense per Discourse* hypothesis proposed by Gale, Church and Yarowsky [22]. Besides the parent phrase head, the part of speech of the parent phrase, is also used as a feature. This is to examine if the part of speech of the parent phrase is useful in partitioning the possible senses of the target word.

2.3.5 Head Word of the Sibling Phrases

Yet another set of phrases which are syntactically related to the phrase which houses the target word are its sibling phrases. As depicted in figure 5, a parent phrase may be made up of many child phrases. We shall call the child phrases *siblings* of each other. Each child may have at most two nearest siblings. A left sibling composed of words immediately preceding the child phrase and a right sibling composed of words immediately following the child phrase. Once again, based on the *One Sense per Discourse* hypothesis, we believe that the head words of these sibling phrases may be suggestive of the topic of discussion and hence indicative of the intended sense of the target word. Consider the instance from Senseval-2 training data shown in Figure 8 (only the sentence housing the target word is shown for brevity, the other sentences of the context being ignored). Figure 11 shows the parsed output of the sentence in Figure 10.

<instance id="nature.40143" docsrc="bnc_A3V_57">
<answer instance="nature.40143" senseid="good_nature%1:07:00::"/>
<context>
With his burly, four-square stance, his ruddy colouring, his handsome strong-featured face,
he radiated energy, warmth and good <head>**nature**</head> .
</context>
</instance>

Figure 10: Head Word of the Left Sibling Phrase: Instance from Senseval-2 Training Data.

Notice that the target word *nature* is part of a noun phrase *good natured*. The left sibling of the noun phrase is another noun phrase *energy, warmth*. The head word of the sibling phrase is the noun *warmth* which along with *energy* and *good nature* are used to describe a face. Since the words are used to describe the same entity, we expect them to be closely related in meaning. Thus, the head word of the left sibling phrase and by symmetry that of the right sibling phrase of the target word may help restrict the possible senses of a word if not uniquely identify it. In this particular instance, the word *warmth* suggests that the target word *nature* has *good nature* sense which corresponds to the sense ID *good_nature%1:07:00::*. Sibling phrase apart, the part of speech of the sibling phrase, is also used as a feature. This is to examine if the part of speech of the sibling phrase is useful in partitioning the possible senses of the target word.

<instance id=”nature.40143” docsrc=”bnc_A3V_57”>
<answer instance=”nature.40143” senseid=”good_nature%1:07:00::”/>
<context>
<P=”P˜radiated˜1˜1”> <P=”S˜radiated˜3˜3”> <P=”PP˜With˜2˜1”> with <p=”IN”/>
<P=”NP-A˜stance˜3˜1”> <P=”NPB˜stance˜4˜4”> his <p=”PRP$”/> burly <p=”JJ”/> , <p=”PUNC,”/>
four-square <p=”NN”/> stance <p=”NN”/> , <p=”PUNC,”/> </P> <P=”NPB˜colouring˜3˜3”> his
<p=”PRP$”/> ruddy <p=”JJ”/> colouring <p=”NN”/> , <p=”PUNC,”/> </P> <P=”NPB˜face˜4˜4”>
his <p=”PRP$”/> handsome <p=”JJ”/> strong - featured <p=”JJ”/> face <p=”NN”/> , <p=”PUNC,”/>
</P> </P> </P> <P=”NPB˜he˜1˜1”> he <p=”PRP”/> </P> <P=”VP˜radiated˜2˜1”> radiated
<p=”VBD”/> <P=”NP-A˜ warmth˜3 ˜1”> <**P=”NPB˜warmth˜2˜2”**> energy <p=”NN”/> ,
<p=”PUNC,”/> **warmth** <p=”NN”/> </P> and <p=”CC”/> <P=”NPB˜nature˜2˜2”> good <p=”JJ”/>
<**head**> **nature** <**/head**> <p=”NN”/> . <p=”PUNC.”/> </P> </P> </P> </P> nature </head>
<p=”NN”/> . <p=”PUNC.”/> </P> </P> </P> </P>
</context>
</instance>

Figure 11: Head Word of the Left Sibling Phrase: Parsed structure of the sentence from Figure 10.

2.4 Resources to Capture the Features

2.4.1 N-gram Statistic Package

Banerjee and Pedersen's N-gram Statistics Package [4] is utilized to capture unigram and bigram features. The features are chosen such that the unigrams and bigrams must occur two or more times in the training data. They must have a log likelihood of 6.635 and they must not have words which are in a pre-decided stop list. The stop list is a file containing a list of non-content words like, *a, the, of, from and at*. The programs count.pl and statistic.pl are utilized with the following options.

```
count.pl --ngram N --extended --newLine --window WINDOW --histogram WORD-training.gram2.hist --nontoken NONTOKEN --token TOKEN --stop STOPLIST WORD-training.gram2.cnt WORD-training.count
```

```
statistic.pl --extended --rank RANK --frequency FREQ --score SCORE log.pm WORD-training.gram2.log WORD-training.gram2.cnt
```

WINDOW is set to 1 for unigrams and 2 for bigrams. A NONTOKEN file is used to eliminate the parse and part of speech XML tags in the files.

2.4.2 The Brill Tagger

Part of speech tagging is a pre-requisite for many Natural Language tasks. Apart from the part of speech of the target word and its surrounding words, syntactically and lexically related words (described in the following sections) may also be used as features for disambiguation. Many chunkers and parsers which are used to obtain these syntactic relations utilize the part of speech information for their functioning. Numerous part of speech taggers such as Ratnaparkhi [65] and QTAG [69] are available commercially and in the public domain. This thesis utilizes the Brill Tagger [8] [9] [10] to part of speech tag the text. The tagger is widely used in the research community as it has the following advantages to its merit.

I: It achieves a tagging accuracy of around 95%

II: Source code of the tagger is available. This enables us to better understand the tagger and use it to the best potential. It allows us to customize it to improve tagging quality based on our requirements.

III: It is based on a transformation based learning. Thus unlike most other taggers which are probabilistic, the information captured by learning is easily understood. In case of the Brill Tagger it learns a set of rules to tag the tokens based on the token itself and its surrounding context.

2.4.3 Collins Parser

Numerous parsers such as the Charniak [14] Parser, MINIPAR [36], Cass Parser [1] [2] and the Collins Parser [16] [37] were considered for use in the thesis. The Collins Parser [16] [37] was selected for the following reasons:

I: Source code of the parser is available. This enables us to better understand the parser and use it to the best potential.

II: It takes as input part of speech tagged text. There are many parsers that take raw sentences as input and both part of speech tag them and provide a parsed structure. This, however, will mean that we shall be unable to utilize the Brill Tagger and guaranteed pre-tagging of the head words which we believe provide high quality part of speech tagging.

III: It has been used widely in the research community to parse text.

As mentioned earlier the Collins Parser takes as input sentences which are part of speech tagged. (13) illustrates the input format for the parser.

`6 Harry NNP Potter NNP cast VBP a DT bewitching JJ spell NN` (13)

A count of the number of words and punctuations in the sentence should be placed at the start of every sentence. sentence 13 has 6 words in all. The word and the part of speech must be separated by white space. The output of the parser is in two formats. A horizontal tree format and a bracketed format. Figure 12 depicts the horizontal tree format output while Figure 13 shows the bracketed format of output from the parser when given the part of speech tagged sentence (13)

This thesis utilizes the bracketed form of the output. Following is description of how one may interpret the parsed output.

```
TOP −33.1527 S −23.6455 NP−A −9.50458 NPB −9.2643 NNP 0 Harry
                                                   NNP 0 Potter
                        VP −11.0818 VBD 0 cast
                                NP−A −6.22721 NPB −5.85218 DT 0 a
                                                    JJ 0 bewitching
                                                    NN 0 spell
```

Figure 12: Sample Horizontal Tree Output of Collins Parser

(TOP~cast~1~1 (S~cast~2~2 (NPB~Potter~2~2 Harry/NNP Potter/NNP)

(VP~cast~2~1 cast/VBD (NPB~spell~3~3 a/DT bewitching/JJ spell/NN))))

Figure 13: Sample Bracketed Output of Collins Parser

I: Token and part of speech separated by forward slash.
Example: Harry/NNP

II: The tree structure is captured by parentheses: (and).

III: Open parenthesis is followed by a non-terminal (NT)
NT Format : NTlabel headword total#ofChildren constituent#
NT Example : NPB spell 3 3 the/DT ball/N

a. *NTlabel* specifies the phrase that follows. In the example above, a noun phrase (NPB) follows.
b. *headword* specifies the head word of the phrase that follows. In the example above, *spell* is the head word the noun phrase that follows.
c. total#ofChildren gives the total number of children of the phrase that follows. The children may be individual words or phrases. In the example above, the noun phrase has 3 children - the words *a, bewitching* and *spell*.
d. *constituent#* specifies the position of the head word in the phrase. In the example above, *spell* is the third child.

2.5 Details of the Brill Tagger and Guaranteed Pre-tagging

2.5.1 The Initial State Tagger

The Brill Tagger part of speech tags sentences in two phases. In the first phase the initial state tagger assigns the most likely part of speech of a word to it. The tagger comes with a LEXICON (LEXICON.BROWN.AND.WSJ) which has a list of around 94,000 words, each followed by its most likely part of speech tag. A list of other possible parts of speech in which the word might exist is also provided. The LEXICON was automatically derived from the Penn TreeBank tagging of the Wall Street Journal and the Brown Corpus. Table 4 depicts a few sample LEXICON entries. Note that there are separate entries for the different surface forms of a word as shown in the table for *football, footballs* and *Football* which are the different surface forms of `football` (L12, L13 and L14).

Table 4: Sample Entries in Brill Tagger's LEXICON

Type	**Most Frequent Tag**	**Other Possible Tags**	
brown	JJ	NN VB	...(L1)
chair	VB	NN	...(L2)
evening	NN	JJ	...(L3)
in	IN	FW NN	...(L4)
meeting	NN	VB	...(L5)
pretty	RB	JJ	...(L6)
sit	VB	FW VB	...(L7)
the	DT	NNP PDT	...(L8)
this	DT	PDT	...(L9)
time	NN	VB	...(L10)
will	MD	VBP NN	...(L11)
football	NN		...(L12)
footballs	NNS		...(L13)
Football	NNP	NN	...(L14)

Consider the the entry L6 for the surface form *pretty*. It specifies that *pretty* exists most frequently as as adverb (RB) but may be an adjective (JJ), as well. L11 indicates that *will* is usually a modal (MD), but may exist as a past participle (VBP) and a noun (NN) in certain sentences. The order of the tags in the *Other Possible Tags* column, in this case MD and NN, is irrelevant and holds no significance. Words which do not occur in the LEXICON will be referred to as words unknown to the tagger or simply unknown words. The initial state tagger assigns the noun or proper noun tag to them based on whether the word is capitalized or not. These unknown words are then subject to another set of rules, based on affixes of the word, which may assign a more suitable part of speech to them. These rules are known as lexical rules. The tagger comes with a pre-defined lexical rule file - LEXICALRULEFILE. The rules have been automatically deduced based on the same corpora from which the LEXICON was learned. The rules have keywords such as *haspref, fhassuf* and *addsuf*. We have categorized the rules based on these keywords. These keywords with example usage and explanation of the rule are listed in Table 5.

2.5.2 Final State Tagger

The part of speech assigned to a word by the initial state tagger depends solely on the word itself. The Final State Tagger may assign a more suitable part of speech to the word based on its context. The context comprises of one to three words to the left and right of the word, along with their parts of speech. It does so by applying a set of contextual rules. The thesis uses the contextual rule file (CONTEXTUALRULEFILE.WSJ) provided in the standard 1.14 distribution of the Brill Tagger. It consists of 284 rules derived from the Penn TreeBank tagging of the Wall Street Journal. Table 6 depicts some sample entries in the contextual rule file.

Here is how we may interpret these rules. C1 specifies that if a word is currently tagged as a noun (NN) and has a determiner (DT) immediately following it, it should be tagged as a verb (VB). C2 tells us that if a word has been assigned the tag of an adverb (RB) but has an noun (NN) immediately following it, its tag is changed to an adjective (JJ). Similarly, as per C3, the tag of word is changed from an adjective to a noun if the next word is a verb. C4 is an example of a rule which utilizes the surface form of a word surrounding the word under consideration to determine its part of speech. Such rules are known as lexicalized contextual rules. C4 tells us that if a word is currently tagged to be a noun (NN) and has the word *meeting* immediately following it, then the tag of the word must be changed to an adjective (JJ).

Table 5: Sample Rules in Brill Tagger's LEXICALRULEFILE

Keyword	**Example Usage**	**Brief Description**
haspref	dis haspref 3 NN x	If a word has the prefix *dis*, which is 3 characters in length, the word is tagged to be a Noun, NN.
fhaspref	VBN un fhaspref 2 JJ x	If a word has prefix *un*, which is 2 characters in length, and is presently tagged as VBN, the word is tagged to be an adjective, JJ.
hassuf	lent hassuf 4 JJ x	If a word has suffix *lent*, which is 4 characters in length, the word is tagged to be an adjective, JJ.
fhassuf	NN ient fhassuf 4 JJ x	If a word has suffix *ient*, which is 4 characters in length, and is presently tagged as a noun NN, the word is tagged to be an adjective, JJ.
char	- char JJ x	If the character '-' appears anywhere in the word, the word is tagged to be an adjective, JJ.
addsuf	ment addsuf 4 VB x	Let X be the word formed by adding the characters *ly* to the word. If X has an entry in the LEXICON, the original word is tagged to be an adjective, JJ.
goodright	Mr. goodright NNP x	If the word appears to the right of the token *Mr.*, the word is tagged to be a proper noun, NNP.
fgoodleft	Co. goodleft NN x	If the word appears to the left of the token *Co.*, the word is tagged to be a proper noun, NNP.

Table 6: Sample Rules in Brill Tagger's CONTEXTUALRULEFILE

Current Tag	**New Tag**	**When**	
NN	VB	NEXTTAG DT	...(C1)
RB	JJ	NEXTTAG NN	...(C2)
JJ	NN	NEXTTAG VB	...(C3)
NN	JJ	NEXTWD meeting	...(C4)

2.5.3 Standard Pre-Tagging with the Brill Tagger

Assigning parts of speech to a subset of all the words in the text before tagging the complete text with a tagger is pre–tagging. We shall refer to such words as pre-tagged words and their assigned parts of speech as pre-tags. Consider the following scenario. It is already known, by an independent means, that the word *chair* in the sentence below is a noun (NN).

`Mona will sit in the pretty chair//NN this time` (14)

Note: All examples in this section have been taken from Mohammad and Pedersen [42].

The word *chair* is pre-tagged to be a noun (NN) to take advantage of this information. When given to the Brill Tagger, the initial state tagger acts first. The words are assigned their most frequent tags based on entries for them in the LEXICON. The parts of speech of the pre-tagged words are not changed. The Initial State Tagger completely ignores the pre-tagged words and does not assign any part of speech to them. Thus, *chair*, continues to have a noun (NN) as the attached part of speech. *Mona* is assigned the tag of a proper noun (NNP) as it does not have an entry in the LEXICON and is capitalized. The output of the Initial State Tagger is as shown below.

`Mona/NNP will/MD sit/VB in/IN the/DT`

`pretty/RB chair//NN this/DT time/NN` (15)

Unlike the Initial State Tagger which treats pre-tagged words differently the Final State Tagger treats both pre-tagged words non-pre-tagged words alike. It may change the tag of the word based on its context. Thus, even though we may pre-tag a word with a certain part of speech, the final state tagger may change the part of speech associated with the pre-tagged word. Pre-tagging is not guaranteed in the standard distribution of the Brill Tagger. This can be seen in the output of the Final State Tagger when given (15):

`Mona/NNP will/MD sit/VB in/IN the/DT`

`pretty/RB chair//VB this/DT time/NN` (16)

We observe that *chair* despite being pre-tagged a noun, is tagged to be a verb (VB) by the Brill Tagger. The change of tag from a noun to verb occurred due to the application of the contextual rule C1 shown in Table 6. The rule specifies that if a word is currently tagged to be a noun and the word immediately following it is a determiner (DT), then the word has been used as a verb (VB). In the sentence above, *chair* is followed by *this* which is a determiner and was pre-tagged to be a noun (NN), thus satisfying the antecedent of the rule and assigned the tag corresponding to a verb (VB). This is an error, since *chair* has been used as a noun. It may also be noted that the error is rippled across to the neighboring word *pretty*. *pretty* is tagged an adverb (RB) by the Initial State Tagger (rule L6 of the LEXICON shown in Table 4). The error in the tagging of *chair* has suppressed the application of rule C2 (Table 6) which would have been applied had *chair* been tagged a noun (NN), changing its tag from an adverb (RB) to an adjective (JJ). The change would have been appropriate as *pretty* is describing the noun *chair*, the trait of an adjective.

In the example above, we note that mis-tagging the head word has caused the suppression of a contextual rule which should have been applied. The converse behavior is also true. The mis-tagging of the pre-tagged words may trigger contextual rules which should not have been applied. Consider the sentence below where, once again, *chair* is pre-tagged to be a noun.

`Mona will sit in the brown chair//NN this time` (17)

The Initial State Tagger does the following assignment of tags based on entries in Table 4.

`Mona/NNP will/MD sit/VB in/IN the/DT`

`brown/JJ chair//NN this/DT time/NN` (18)

Assignment of part of speech tags by the Final State Tagger:

`Mona/NNP will/MD sit/VB in/IN the/DT`

`brown/NN chair//VB this/DT time/NN` (19)

We observe that once again, the pre–tag of *chair* is over-ridden by the Final State Tagger which has assigned the verb (VB) tag to it by the application of contextual rule C1. Unlike the previous example where the mis-tagging of the pre-tagged word lead to the suppression of a contextual rule, this error triggers the rule C3 changing the tag of *brown* from adjective (JJ) to noun (NN). Since *brown* is used as an adjective in this sentence, we note that mis-tagging the pre-tagged word has yet again lead to an erroneous part of speech being assigned to its neighbor.

To summarize, the standard distribution of the Brill Tagger allows pre-tagging but may change the pre-tag based on contextual rules. This transformation will mean that the pre-tag will not affect the selection of part of speech of its neighbors and the mis-tag may influence the tagging of its neighbors. All these aspects are undesirable as they may lead to erroneous part of speech tagging. The next section details how we overcome these drawbacks of the Brill Tagger.

2.5.4 Guaranteed Pre–Tagging

We believe that if the part of speech of a word is known prior to tagging and it is pre-tagged the same, the pre-tag should be respected all through the tagging process and not be changed to another tag. As shown in the previous section, mis-tagging the pre-tagged words may lead to erroneous part of speech assignments to its neighbors as well. Brill Tagger's Initial State Tagger conforms to this requirement but not the Final State Tagger. We have devised a way to guarantee pre-tagging [42] by the Brill Tagger. A patch to the Brill Tagger called the `BrillPatch` [41] is created which makes a simple change in the working of the Final State Tagger, which is as follows: the application of contextual rules to a pre-tagged word is suppressed while continuing to allow the application of contextual rules to all other words. Thus, the Final State Tagger does not change the tag of a pre-tagged word and uses this pre-tag in the selection of appropriate part of speech tags for its neighbors. Consider the example sentences 8 and 17. We observed in the previous section that even though both sentences had the *chair* tagged as a noun the standard distribution of the Brill Tagger, tagged them to be verbs (VB). Further, due to this error, both *pretty* and *brown* were erroneously tagged. With guaranteed pre–tagging the output of the initial state tagger remains the same. The output of the Final State Tagger is as follows:

`Mona/NNP will/MD sit/VB in/IN the/DT`

`pretty/JJ chair//NN this/DT time/NN` (20)

Note: All examples in this section have been taken from Mohammad and Pedersen [42].

We observe that *chair* continues to have noun (NN) as the assigned part of speech which is the what it was pre-tagged to. The contextual rule C1 is suppressed when tagging the pre-tagged words *chair*, due to the modification to the Final State Tagger allowing the pre-tag to remain. As *chair* remains a noun, contextual rule C2 is applied to *pretty* changing its tag from an adverb (RB) to an adjective (JJ).

Sentence 17 is assigned the following parts of speech by the patched Brill Tagger:

`Mona/NNP will/MD sit/VB in/IN the/DT`

`brown/JJ chair//NN this/DT time/NN` (21)

Once again, the pre–tag of *chair* has not been changed. Further, the mis-triggering of contextual rule C2 changing the tag of *brown* from adjective (JJ) to adverb (RB) does not occur.

Thus, guaranteed pre-tagging meets our requirements of allowing the pre-tag to remain and suitably affect the selection of tags of its neighbors.

2.6 An Optimal Subset of Features

Ideally, we would like to use as many features as possible to represent the data, so that we utilize most of the available information. However, tools are needed to capture the features and the process of capturing the features might be costly in terms of time and effort. Secondly, the larger the feature set, the longer the time taken to learn the classifier and greater the danger of fragmentation. Since, a classifier is to be learnt for every word to be disambiguated, the time factor becomes significant. Capturing a multitude of features from large data sets and learning from them, in a reasonable amount of time thus poses a problem.

Another driving force towards choosing a subset of the possible features is that many of the features appear to possess the same discriminating information. For example consider the features formed using part of speech of a word and its surface form. Table 7 shows the various surface forms and possible parts of speech of the verb *take*.

Table 7: Surface Form - Part of Speech Correspondence.

Surface Form	**Part of Speech**
take	VB (verb base form)
take	VBP (verb present)
takes	VBZ (verb present, 3d person)
took	VBD (verb past)
taking	VBG (gerund)
taken	VBN (past participle)

Notice, that there is an almost one to one correspondence between them, suggesting that the information gained by the exact part of speech may not help much more than surface form. The instances correctly classified by these features independently may have significant overlap i,e, a number of instances appropriately tagged by a classifier which uses part of speech of head word, may be correctly tagged by a classifier learnt from surface form, as well. Of course, the same is not so much the case for other parts of speech. We believe that there might be other such pairs of knowledge sources as well, where one knowledge source helps correctly classify, more or less, the same instances as the other. Albeit, it might not be as evident as in the case of surface form and part of speech. On the other hand, pairs of knowledge sources are good at classifying significantly complementary sets of instances are also likely to exist. We would thus like to use a minimal set of features for disambiguation without much loss of accuracy. Learning the amount of redundancy and complementarity amongst the various features will help in making tradeoffs between accuracy and cost.

In this thesis we have concluded that the syntactic and lexical features are significantly complementary. We show that the decision tree created by the combination of the target word part of speech, and the parts of speech of its two adjacent words performs best in combination with the lexical features unigrams and bigrams, as opposed to the other combinations of part of speech features. We will show that the part of speech of the word to the right of the target word is the most useful feature for sense disambiguation amongst all the individual word part of speech features. We find that nouns benefit from part of speech tags on its either side while verbs and adjectives are disambiguated better using the part of speech tags of words to their immediate right. We show that the head word of a phrase is particularly useful to disambiguate adjectives. The head of the phrase and the head of the parent phrase have proved to be useful for nouns. We show that guaranteed part of speech tagging, which was employed in the part of speech tagging of all the data, has helped word sense disambiguation.

3 EXPERIMENTAL DATA

3.1 Sense Tagged Corpora

Sense tagged data used to train the classifier is created by manual annotation. A person or a team of people tag each target word with a sense appropriate for that instance. Such manually annotated data is of much better quality than that amassed through automatic or semi-automatic techniques. However, its creation is both expensive and time intensive. The Senseval-1 and Senseval-2 exercises, held in the summers of 1998 and 2001 respectively, brought together numerous word sense disambiguation systems from all over the world to carry out a common evaluation. All the systems were trained and evaluated on a common set of sense tagged data set created specifically for these exercises. Senseval-1 data has 13,276 sense tagged instances as training data and 8,452 test instances, ranging over 35 nouns, verbs and adjectives. Senseval-2 data has 8,611 training and 4,328 test instances covering 73 nouns verbs and adjectives. To date, these corpora boast of being the largest repositories of sense tagged data. Earlier attempts at creating sense tagged data were concentrated on specific words. The *line, hard, serve* and *interest* data have considerable amounts of annotated data corresponding to the respective words. The *line* data, created by Leacock [31], has 4,149 sense tagged instances of the noun *line*. The *hard* and *serve* data, created by Leacock, Chodorow and Miller [31], have 4,337 and 4,378 instances of the adjective *hard* and the verb *serve*, respectively. The *interest* data which has 2,476 instances of the noun *interest* was created by Bruce and Wiebe [13].

Systems which utilize sense tagged data place their own requirements on the format of the data. An offshoot of the Senseval-1 and Senseval-2 exercises was that all the systems taking part were designed to accept a common data format. However, Senseval-1 and Senseval-2 data were in different formats. Since the creation of new manually annotated data is expensive, it is much more viable to convert the available sense tagged data into a common format. We chose Senseval-1 and Senseval-2 data formats as much of the present and future work is being done in them. As a part of this thesis, packages are provided to convert the *line*, *hard*, *serve* and *interest* data into Senseval-1 and Senseval-2 data format. These packages use `Sval1to2` [3] to covert the data from Senseval-1 to Senseval-2 data format. The `Sval2Check` [51] package was developed to check if the data after conversion to Senseval-2 data format and the original Senseval-2 data, for that matter, does indeed conform to Senseval-2 data format. The package also checks for duplicate instance IDs and contexts. Due to these packages, sense tagged data which would otherwise be unusable to many systems due to their formats, may now be used by all the systems which accept data in the Senseval-1 or Senseval-2 data format. The *line*, *hard*, *serve*, *interest*, Senseval-1 and Senseval-2 data, with over 50 thousand sense tagged instances covering 112 words, form a significant set of data for our experiments. Details of individual sources of data follow.

3.1.1 *line* Data

The *line* data was created by Leacock et. al. [32] and consists of 4,149 instances that have 2 or 3 lines of text one of which contains the noun *line* as the target word. The instances were picked from the Wall Street Journal Corpus (1987-89) and the the American Printing House for the Blind (APHB) corpus. Each instance was manually annotated with one of six senses of the noun *line* from WordNet. The data is provided in six files; one corresponding to every sense. A sample instance from the *line* data is provided in Figure 14. We developed the `lineOneTwo` [46] package which converts data in *line* data format to Senseval-1 data format. The package then uses `Sval1to2` [3] to convert it from Senseval-1 to Senseval-2 data format. The sample instance in Senseval-1 and Senseval-2 data formats is shown in Figures 15 and 16. The sense id assigned to the instances and the distribution of instances is as shown in Table 8. A description of the senses with example usages of *line* in these senses is provided in Table 9.

w7_010:888: <s> The company argued that its foreman needn't have told
the worker not to move the plank to which his lifeline was tied because
"that comes with common sense." </s> </p> @ <p> @ <s> The
commission noted, however, that Dellovade hadn't instructed its employees
on how to secure their lifelines and didn't heed a federal inspector's earlier
suggestion that the company install special safety lines inside the A-frame
structure it was building. </s>

Figure 14: A sample instance from the *line* Data.

Since there exist instances with multiple sentences such that the the individual sentences are not placed one per line, a simple sentence detector (part of the package) is used to do the same. The data does not have the w7_010:888:

<s> The company argued that its foreman needn't have told the worker not to move the plank to which his lifeline was tied because "that comes with common sense. " </s> <@> </p> <@> <p> <@> <s> The commission noted, however, that Dellovade hadn't instructed its employees on how to secure their lifelines and didn't heed a federal inspector's earlier suggestion that the company install special safety <tag "cord">lines</> inside the A-frame structure it was building. </s>

Figure 15: The instance from Figure 14 in Senseval-1 data format.

<lexelt item="line-n">
<instance id="line-n.w7_010:888:">
<answer instance="line-n.w7_010:888:" senseid="cord"/>
<context>
<s> The company argued that its foreman needn't have told the worker not to move the plank to which his lifeline was tied because "that comes with common sense. " </s> <@> </p> <@> <p> <@> <s> The commission noted, however, that Dellovade hadn't instructed its employees on how to secure their lifelines and didn't heed a federal inspector's earlier suggestion that the company install special safety <head>lines</head> inside the A-frame structure it was building. </s>
</context>
</instance>
</lexelt>

Figure 16: The instance from Figure 14 in Senseval-2 data format.

Table 8: Senses of *line* and Instance Distribution

Sense	**Sense-Id**	**File**	**No. of Instances**	**Percentage**
Product	product2	product2	2218	53.5%
Phone	phone2	phone2	429	10.3%
Text	text2	text2	404	09.7%
Division	division2	division2	376	09.1%
Cord	cord2	cord2	373	09.0%
Formation	formation2	formation2	349	08.4%
TOTAL			**4149**	**100.0%**

Table 9: Brief Meaning and Example Usages of the Senses of *line*

Sense	**Meaning**	**Example Usage**
Product	a product	a new *line* of mid-sized cars
Phone	a telephone connection	the toll-free help *line*
Text	spoken or written text	one winning *line* from that speech
Division	an abstract division	draw no *line* between work and religion
Cord	a thin, flexible object	a *line* tied to his foot
Formation	a formation of people or things	people waited patiently in long *lines*

target word *line* explicitly marked. The first occurrence of any of the following surface forms of *line* in an instance, is treated as the target word - *line, lines, Line* and *Lines*. Table 10 depicts the proportion of such instances in the different senses. Random spot checks of such instances have indicated that all occurrences of the noun *line* in the same instance have the same sense hence picking any one of them as the head word would have been fine. Gale, Church and Yarowsky's [22] 'One Sense Per Discourse' is also in support of this assumption.

Table 10: Distribution of instances with multiple possible target words in *line* data.

Sense	Multiple Target Words	Total Instances	Percentage
Cord	36	373	9.7%
Phone	35	429	8.2%
Product	126	2218	5.7%
Formation	15	349	4.3%
Text	17	404	4.2%
Division	13	376	3.5%
TOTAL	**242**	**4149**	**5.8%**

It should be noted that the instance with ID *w7_089:15499:* is garbled and hence ignored by the package. The number of instances of the *product* file converted to Senseval-1 and Senseval-2 data formats is thus 2,217. Additionally, the original data has duplicate instances (instances with the same instance ID and context) in the *division* file, corresponding to the instance IDs - w8_055:13056: and w8_056:9116:. These duplicates have been removed. Thus, the division file in Senseval-1 and Senseval-2 data formats has 374 instances. Due to the removal of these three instances, in all, 4,146 instances of *line* data have been converted to Senseval-2 data format and used in our experiments. The converted data was validated to be in correct Senseval-2 data format without any other duplicate instance IDs or contexts, by the `Sval2Check` package.

Since its creation, the *line* data has been used to evaluate numerous word sense disambiguation systems. Leacock, Towell and Voorhees [32] evaluated their sense resolution system, which used both local context and the broad topic of discussion, on this data. Pedersen [54] tested his system, an ensemble of Naive Bayesian classifiers based on co-occurring words, on the *line* data. The data was used to study the effectiveness of EM algorithm and Gibb's sampling method to extract sense discrimination knowledge from untagged text by Pedersen [58]. Leacock, Chodorow and Miller [31] applied their sense disambiguation system which used relations from WordNet, on *line* data.

3.1.2 *hard* Data

The *hard* data was created by Leacock, Chodorow and Miller [31]. It consists of 4,337 instances, each with a single sentence that has the adjective *hard* as the target word. The instances were picked from the San Jose Mercury News Corpus (SJM) and manually annotated with one of three senses from WordNet. The data is provided in three files; one corresponding to each sense. A sample instance of the *hard* data is depicted in Figure 17. We developed the `hardOneTwo` [44] package which converts data in *hard* data format to Senseval-1 data format. The package then uses Sval1to2 [3] to convert it from Senseval-1 to Senseval-2 data format. The sample instance in Senseval-1 and Senseval-2 data formats is shown in Figures 18 and 19. The sense id assigned to the instances and the distribution of instances is as shown in Table 11. A description of the senses with example usages of *hard* in these senses is provided in Table 12.

sjm-274:" He may lose all popular support , but someone has to kill him
to defeat him and that 's HARD to do. "

Figure 17: A sample instance from the *hard* Data.

sjm-274:
<s> " He may lose all popular support , but someone has to kill him to
defeat him and that 's <tag "HARD1">HARD</> to do. " </s>

Figure 18: The instance from Figure 17 in Senseval-1 data format.

The data does not have the target word explicitly marked. The first occurrence of any of the following surface forms of *hard* in an instance, is treated as the target word - *hard, harder, hardest, HARD, HARDER* and *HARDEST*. There are instances with two of these surface forms. Table 13 depicts the proportion of such instances in the different senses. Of all the surface forms mentioned here, *hard, harder, HARD, HARDER* and *HARDEST* exist in the *hard* data. The package, however, supports the conversion of any data in the *hard* data format to Senseval-1 and Senseval-2 data formats even if it has *hardest* as the head word. It may

<lexelt item=”hard-a”>

<instance id=”hard-a.sjm-274:”>

<answer instance=”hard-a.sjm-274:” senseid=”HARD1”/>

<context>

<s> “ He may lose all popular support , but someone has to kill him to
defeat him and that ’s <head>HARD</head> to do. ” </s>

</context>

</instance>

</lexelt>

Figure 19: The instance from Figure 17 in Senseval-2 data format.

Table 11: Senses of *hard* and Instance Distribution

Sense-Id	**File**	**No. of Instances**	**Percentage**
HARD1	hard1.A	3455	79.7%
HARD2	hard2.A	502	11.6%
HARD3	hard3.A	380	08.7%
TOTAL		**4337**	**100.0%**

Table 12: Brief Meaning and Example Usages of the Senses of *hard*

Sense ID	**Meaning**	**Example Usage**
HARD1	Not easy - difficult	its *hard* to be disciplined
HARD2	Not soft - metaphoric	these are *hard* times
HARD3	Not soft - physical	the *hard* crust

be noted that there is an occurrence of the type *HARDplastic* in the instance *sjm-074:*. This token is broken into two tokens, *HARD* and *plastic*, which is more likely intended.

The converted data was validated to be in correct Senseval-2 data format by the `Sval2Check` [51] package. However, it was found to have four pairs of true duplicate instances - instances with the same instance ID, context and sense ID. These instances correspond to instance IDs *sjm-206:, sjm-212:, sjm-219:* and *sjm-221:* in the `hard1.A` file. These duplicates were removed from the data. Thus, there are 3451 instances corresponding to the HARD1 sense and 4333 instances in all. It was also found that the 349 instance IDs were being used in 4,186 instances with different contexts, meaning they could not be used as unique identifiers. Since, instance IDs are to have a one on one mapping with the contexts, we developed a perl program `unique-hard.pl` [52] which transformed the instance IDs of the 4,186 instances to 4,186 mutually distinct strings. If there are 4 instances with the instance ID *sjm-274:*, the program gives them the IDs *sjm-274_1:, sjm-274_2, sjm-274_3* and *sjm-274_4*, in the order that the instances are found in the file. The data in Senseval-2 data format created from this pre-processed *hard* data was validated to have no duplicate instances or contexts by `Sval2Check`. It was also found to be in correct Senseval-2 data format. *hard* data like *line* was used to evaluate Leacock, Chodorow, and Miller's [31] sense resolution system based on relations from WordNet.

Table 13: Distribution of instances with multiple possible target word in *hard* data.

Sense	Multiple Target Words	Total Instances	Percentage
HARD1	8	3455	0.2%
HARD2	3	502	0.6%
HARD3	2	380	0.5%
TOTAL	**13**	**4337**	**0.3%**

3.1.3 *serve* Data

The *serve* data was also created by Leacock, Chodorow and Miller [31]. It consists of 5,131 instances, with two to three sentences, that have the verb *serve* as the target word. The instances, like *line* data, were picked from the Wall Street Journal Corpus (1987-89) and the the American Printing House for the Blind (APHB) corpus. They have been manually annotated with four senses from WordNet. A sample instance of the *serve* data is depicted in Figure 20. The instances are tagged with one of the four senses of *serve*. The data is provided in four files; one corresponding to every sense. We developed the `serveOneTwo` [50] package which converts data in *serve* data format to Senseval-1 data format. The package then uses Sval1to2 [3] to convert it from Senseval-1 to Senseval-2 data format. The sample instance in Senseval-1 and Senseval-2 data formats is shown in Figures 21 and 22. The sense ID assigned to the instances and the distribution of instances is as shown in Table 14. A description of the senses with example usages of *serve* in these senses is provided in Table 15.

aphb_09701001_665 The agreement was to be kept secret . Paine saw no objection
to being paid for writing in this vein , but the affair of the Indiana Company had
served as a warning that his motives might not be understood .

Figure 20: A sample instance from the *serve* Data.

aphb_09701001_665
<s> The agreement was to be kept secret . </> <s> Paine saw no objection
to being paid for writing in this vein , but the affair of the Indiana Company had
<tag "SERVE2">served</> as a warning that his motives might not be understood . </s>

Figure 21: The instance from Figure 20 in Senseval-1 data format.

Table 14: Senses of *serve* and Instance Distribution

Sense-Id	**File**	**No. of Instances**	**Percentage**
SERVE10	serve10.A	1814	41.4%
SERVE12	serve12.A	1272	29.1%
SERVE2	serve2.A	853	19.5%
SERVE6	serve6.A	439	10.0%
TOTAL		**4378**	**100.0%**

```
<lexelt item=”serve-v”>
<instance id=”serve-v.aphb_09701001_665”>
<answer instance=”serve-v.aphb_09701001_665” senseid=”SERVE2”/>
<context>
<s> The agreement was to be kept secret . </> <s> Paine saw no objection
to being paid for writing in this vein , but the affair of the Indiana Company had
<head>served</head> as a warning that his motives might not be understood . </s>
</context>
</instance>
</lexelt>
```

Figure 22: The instance from Figure 20 in Senseval-2 data format.

Table 15: Brief Meaning and Example Usages of the Senses of *serve*

Sense ID	Meaning	Example Usage
SERVE2	Supply with food/means	this instrument serves two purposes
SERVE6	Hold an office	the department will now *serve* a select few
SERVE10	Function as something	he *served* as the chief inspector
SERVE12	Provide a service	selected to *serve* on a destroyer

Since, the instances are composed of multiple sentences, such that individual sentences are not placed on new lines, a simple sentence detector (part of the package) is used to place sentences on new lines. The data does not have the target word explicitly marked. The first occurrence of any of the following surface forms of *serve* in an instance is treated as the target word - *serve*, *served*, *serves*, *serving, Serve, Served, Serves* and *Serving*. Table 17 depicts the proportion of such instances in the different senses. Of all the forms mentioned here, *serve, served, serves, Serve* and *Serves* exist in the *serve* data. The package, however, supports the conversion of any data in the *serve* data format to Senseval-1 and Senseval-2 data formats even if it has *Served* as the head word. The converted data was validated to be in correct Senseval-2 data format by the `Sval2Check` package. The program flagged nine pairs of instances (listed in Table 16) to have the same context and sense ID but differing instance IDs. This is a kind of duplication and gives a little more weightage to getting these contexts right but not an error in any way. Due to the small number, these instances have been left as is. No other duplicates were found. The *serve* data, like *line* and *hard* was used to evaluate Leacock, Chodorow, and Miller's [31] sense resolution system based on relations from WordNet.

Table 16: *serve*: same context and sense IDs but different instance IDs. Not removed.

.

Instance 1	**Instance 2**	**Sense**
serve-v.aphb_39400043_2779	serve-v.aphb_21000043_1310	SERVE10
serve-v.aphb_37501758_2485	serve-v.aphb_37502314_2541	SERVE10
serve-v.aphb_37501389_2448	serve-v.aphb_37501085_2416	SERVE10
serve-v.aphb_37501418_2452	serve-v.aphb_37501069_2414	SERVE10
serve-v.aphb_37501594_2469	serve-v.aphb_37502937_2619	SERVE10
serve-v.aphb_37500603_2359	serve-v.aphb_37502937_2619	SERVE10
serve-v.aphb_37502233_2532	serve-v.aphb_37501174_2424	SERVE10
serve-v.aphb_08800006_600	serve-v.aphb_08800035_601	SERVE12
serve-v.w7_0194875_454	serve-v.w7_02012979_487	SERVE12

Table 17: Distribution of instances with multiple possible target words in *serve* data.

Sense	Multiple Target Words	Total Instances	Percentage
SERVE10	275	1814	15.2%
SERVE6	19	439	4.3%
SERVE12	48	1272	3.8%
SERVE2	10	853	1.2%
TOTAL	**352**	**4378**	**8.0%**

3.1.4 *interest* Data

The *interest* data was created by Bruce and Wiebe [13]. It consists of 2,368 instances. Each with one sentence that has the noun *interest* as the target word. The instances have been selected from the Penn Treebank Wall Street Journal corpus (ACL/DCI version) and manually annotated with one of the six senses from the Longman Dictionary of Contemporary English (LDOCE) [59]. Specifically, the instances were taken from the parsed subset of the corpus. The tokens are tagged with their parts of speech and the parse information of the sentence is also provided via appropriate bracketing. The data is provided in a single file. A sample instance of the *interest* data is depicted in Figure 23. Since, we would like to use the Brill Tagger and Collins parser [16] [37] to part of speech tag and parse the data, a version of the *interest* data without the part of speech tags and parse information has been created. A sample of the instance without these tags is shown in Figure 24. We developed the `interestOneTwo` [45] package which converts data in *interest* data format to Senseval-1 data format. The package then uses Sval1to2 [3] to convert it from Senseval-1 to Senseval-2 data format. The sample instance in Senseval-1 and Senseval-2 data formats is shown in Figures 25 and 26. The sense ID assigned to the instances and the distribution of instances is as shown in Table 18. A description of the senses with example usages of *interest* in these senses is provided in Table 19.

The data has the target word explicitly marked with an underscore and the sense number immediately following the head word. For example *interests_5*, where *interests* is the head word and *5* is the sense number corresponding to this instance. The surface forms of *interest* which may be considered as head words are

[yields/NNS] on/IN [money-market/JJ mutual/JJ funds/NNS] continued/VBD to/TO
slide/VB ,/, amid/IN [signs/NNS] that/IN [portfolio/NN managers/NNS] expect
/VBP [further/JJ declines/NNS] in/IN [interest_6/NN rates/NNS] ./.

Figure 23: A sample instance from the *interest* Data.

yields on money-market mutual funds continued to slide , amid signs that
portfolio managers expect further declines in interest_6 rates .

Figure 24: The sample instance of *interest* data without the part of speech and parse tags.

int1
<s> yields on money-market mutual funds continued to slide , amid signs
that portfolio managers expect further declines in <tag "interest_6">
interest </> rates . </s>

Figure 25: The instance from Figure 24 in Senseval-1 data format.

<lexelt item="interest-n">
<instance id="interest-n.int1">
<answer instance="interest-n.int1" senseid="interest_6"/>
<context>
<s> yields on money-market mutual funds continued to slide , amid signs
that portfolio managers expect further declines in <head>interest</head>
rates . </s>
</context>
</instance>
</lexelt>

Figure 26: The instance from Figure 24 in Senseval-2 data format.

Table 18: Senses of *interest* and Instance Distribution

Sense-Id	**File**	**No. of Instances**	**Percentage**
interest_6	interest.acl94.txt	1252	52.9%
interest_5	interest.acl94.txt	500	21.1%
interest_1	interest.acl94.txt	361	15.2%
interest_4	interest.acl94.txt	178	07.5%
interest_3	interest.acl94.txt	66	02.8%
interest_2	interest.acl94.txt	11	00.5%
TOTAL		**2368**	**100.0%**

Table 19: Brief Meaning and Example Usages of the Senses of *interest*

Sense ID	**Meaning**	**Example Usage**
interest_1	Readiness to give attention	international *interest* in Iraq
interest_2	Quality of causing attention to be given to	video games may be of *interest*
interest_3	Activity, etc. that one gives attention to	pursue other *interests*
interest_4	Advantage, advancement or favor	in best *interest* of my client
interest_5	A share in a company or business	the company has *interests* in real estate
interest_6	Money paid for the use of money	higher *interest* rates

Table 20: *interest*: same context and sense IDs but different instance IDs. Not removed.

Instance 1	**Instance 2**	**Sense**
interest-n.int2030	interest-n.int2059	interest_5
interest-n.int628	interest-n.int2053	interest_5
interest-n.int627	interest-n.int2052	interest_5
interest-n.int626	interest-n.int2051	interest_5
interest-n.int937	interest-n.int1535	interest_6
interest-n.int1422	interest-n.int2176	interest_6
interest-n.int1899	interest-n.int2254	interest_6
interest-n.int161	interest-n.int1899	interest_6
interest-n.int161	interest-n.int1548	interest_6
interest-n.int161	interest-n.int500	interest_6
interest-n.int161	interest-n.int2254	interest_6
interest-n.int1548	interest-n.int1899	interest_6
interest-n.int1548	interest-n.int2254	interest_6
interest-n.int500	interest-n.int1899	interest_6
interest-n.int500	interest-n.int1548	interest_6
interest-n.int500	interest-n.int2254	interest_6
interest-n.int105	interest-n.int106	interest_6
interest-n.int80	interest-n.int483	interest_6
interest-n.int2004	interest-n.int2005	interest_6
interest-n.int936	interest-n.int1534	interest_6
interest-n.int361	interest-n.int1420	interest_6
interest-n.int361	interest-n.int1136	interest_6
interest-n.int1136	interest-n.int1420	interest_6

interest and *interests*. The converted data was validated to be in correct Senseval-2 data format by the `Sval2Check` package. However, like *serve* data, the program flagged 23 pairs of instances (listed in Table 20 to have the same context and sense ID but differing instance IDs. This is a kind of duplication and gives a little more weightage to getting such instances right but not an error in any way. Due to the small number, these instances have been left as is. No other duplicates were found. Bruce and Wiebe [13] performed a case study on *interest* data to evaluate their probabilistic disambiguation system based on multiple contextual features.

3.1.5 Senseval-1 English Lexical Sample Task

The Senseval-1 [28] [29] exercise was conducted in the summer of 1998. Systems were tested on English, French and Italian corpora. Within each language, a further subdivision was made based on whether the system would attempt disambiguation of all words in the text or specific ones. The English Lexical Sample Task corresponds to the latter and is what our disambiguation system does. Unlike, the *line*, *hard*, *serve* and *interest* data which are not divided into a test and training corpus, the Senseval-1 data, as provided, exists as separate test and training corpora. The test corpus has 8,512 instances and the training corpus has 13,276. Hector, the dictionary built along with the corpora acts as their sense inventory. Each instance is composed of two to three sentences. A sample instance from Senseval-1 test and training data is shown in Figures 27 and 28, respectively.

800122
Their influences include the Stones and Aerosmith but I thought the
track Hell's Kitchen had overtones of Tom Petty and Tom Verlaine.
Stage Dolls &dash. Stage Dolls (Polydor) Hailed as Norway's premier
rock <tag "532736">band</>, they have supported Michael Monroe.

Figure 27: A sample training instance from Senseval-1 data.

Sense tagged data for thirty six nouns, verbs and adjectives of the English language are provided. Seven of the words have instances corresponding to both noun and verb form. The distribution of these instances in the test and training data as per part of speech, along with the number of possible senses for each task is

700003
It is obviously a very exciting project for us though, and it will
allow Kylie to show just how much she has developed over the past
three years. 'It still <tag>amazes</> me how much she has
come on and is improving all the time.

Figure 28: A sample test instance from Senseval-1 data.

listed in Table 21. The number of senses is as found in the training data. Instances corresponding to another five words have also been provided, but the part of speech of the target word in these instances is not known. These tasks are referred as indeterminates. The distribution of the indeterminates is listed in Table 22. It may be noted that five of the words (marked in bold in the tables) - *disability, rabbit, steering, deaf* and *hurdle* which have test data, do not have any training data. Thirteen of the words, italicized in the tables, have a very small number of training examples and no test examples. The instances corresponding to these words have no bearing on our experiments.

Senseval-1 data has certain mis-tagged sentences. For example, sentences where *Silver* is tagged as the head word, albeit the head word should have been band (see example instance in Figure 29. Other such examples include brass band, *big band* and *hand shake*. There are four such instances in the test data and 62 in the training data. We developed the `Senseval1-fix` [43] package to correct these bugs. Figure 30 shows the corrected form of the example instance. The data is converted to Senseval-2 data format using the `Sval1to2` [3] package. The sample instances in Senseval-2 data format are shown in Figures 31 and 32, respectively. The converted data was validated to be in correct Senseval-2 data format by the `Sval2Check` [51] package. However, like *serve* and *interest* data, the program flagged 10 pairs of instances (listed in Table 23) from the test data and 14 pairs of instances (listed in Table 24) to have the same context and sense ID (in case of training data) but differing instance IDs. This is a kind of duplication and gives a little more weightage to getting such instances right but not an error in any way. Due to the small number, these instances have been left as is. Additionally, the training data had two pairs of true duplicates i,e, same context, same instance ID and the same sense IDs. These correspond to the instances *float-v.800287* and *float-v.800297*. The training data was also found to have 131 pairs of instances with the same instance ID, same context and differing sense IDs. This is due to the way `Sval1to2` handles instances with multiple senses. Given an instance in Senseval-1 data format, with three senses, `Sval1to2` [3] creates three

Table 21: Senseval-1: Instance Distribution of Nouns, Verbs and Adjectives

Nouns	Count		Senses	Verbs	Count		Senses	Adjectives	Count		Senses
	Test	Train			Test	Train			Test	Train	
accident	267	1238	8	amaze	70	133	1	brilliant	229	443	11
behavior	279	998	3	bet	178	67	7	**deaf**	123	–	–
bet	275	110	10	bother	209	282	8	floating	47	42	5
disability	160	–	–	bury	201	290	12	generous	227	308	6
excess	186	178	8	calculate	218	219	5	giant	97	317	6
float	75	63	8	consume	186	61	6	modest	270	383	9
giant	118	344	8	derive	217	266	7	slight	218	380	6
knee	251	417	6	float	229	200	15	wooden	196	362	5
onion	214	26	2	invade	207	49	6	*amaze*	–	183	1
promise	113	589	9	promise	224	1175	7	*calculate*	–	31	2
rabbit	221	–	–	sack	178	187	3	*consume*	–	11	1
sack	82	99	7	scrap	186	30	2	*excess*	–	73	1
scrap	156	27	9	seize	259	290	11	*invade*	–	8	2
shirt	184	531	8	*knee*	–	2	1	*knee*	–	16	6
steering	176	–	–					*promise*	–	262	3
bother	–	12	4					*seize*	–	4	3
brilliant	–	2	2					*shirt*	–	2	1
slight	–	5	2								
TOTAL	**2757**	**4639**	**94**	**TOTAL**	**2562**	**3251**	**91**	**TOTAL**	**1407**	**2825**	**67**

Table 22: Senseval-1: Instance Distribution of Indeterminates

Indeterminates	**Count**		**No. of Senses**
	Test	**Train**	
band	302	1330	24
bitter	374	144	11
hurdle	323	–	–
sanction	431	96	5
shake	356	991	30
TOTAL	**1786**	**2561**	**70**

instances in Senseval-2 data format corresponding to it. They have the same instance ID and context but one different sense each. `Sval2Check` [51] flagged 19 special cases (Table 25) in the training data. Each special case corresponds to a pair of instances with the same context, different instance ID and different sense ID. These are cases of true part of speech ambiguity, where the part of speech of the target word could be at least two different tags for the same context. No other duplicates were found.

800123
(Mrs) BONNIE WORCH Clements Green, South Moreton, Didcot.
THE Wantage **<tag ”532747-p”>Silver</> Band** collected all
the trophies in Section C at the Oxford and District Brass
Band Association contest in Oxford recently.

Figure 29: A sample instance from Senseval-1 data which has an erroneous target word tag.

Yarowsky [75] implemented a word sense disambiguation system using hierarchical decision lists and a rich set of features. The system took part in the Senseval-1 exercise held in the summer of 1998 and achieved and accuracy of 78.4%. Lee and Ng [33] performed sense disambiguation experiments with a number of knowledge sources and supervised learning algorithms. Senseval-1 data was used to evaluate the performance of the various systems. Florian and Yarowsky [21] studied the combination of classifiers to

800123
(Mrs) BONNIE WORCH Clements Green, South Moreton, Didcot.
THE Wantage **Silver** <**tag ”532747-p”**>**Band**</> collected all
the trophies in Section C at the Oxford and District Brass
Band Association contest in Oxford recently.

Figure 30: he sample instance from Senseval-1 data which has been corrected by `Senseval1-fix`.

<lexelt item=”band-p”>
<instance id=”band-p.800122”>
<answer instance=”band-p.800122” senseid=”532736”/>
<context>
Their influences include the Stones and Aerosmith but I thought the
track Hell’s Kitchen had overtones of Tom Petty and Tom Verlaine.
Stage Dolls &dash. Stage Dolls (Polydor) Hailed as Norway’s premier
rock <head>band</head>, they have supported Michael Monroe.
</context>
</instance>
</lexelt>

Figure 31: The Senseval-1 training instance in Senseval-2 data format.

<lexelt item=”amaze-a”>
<instance id=”amaze-a.700003”>
<context>
It is obviously a very exciting project for us though, and it will
allow Kylie to show just how much she has developed over the past
three years. ‘It still <head>amazes</head> me how much she has
come on and is improving all the time.
</context>
</instance>
</lexelt>

Figure 32: The Senseval-1 Test Instance in Senseval-2 data format.

Table 23: Senseval-1 test: same context and sense IDs but different instance IDs. Not removed.

Task	Instance 1	Instance 2
disability-n	disability-n.700001	disability-n.700159
float-v	float-v.700198	float-v.700318
float-v	float-v.700133	float-v.700299
float-v	float-v.700211	float-v.700216
hurdle-p	hurdle-p.700158	hurdle-p.700307
invade-v	invade-v.700070	invade-v.700148
invade-v	invade-v.700036	invade-v.700088
sanction-p	sanction-p.700023	sanction-p.700225
sanction-p	sanction-p.700345	sanction-p.700352
seize-v	seize-v.700019	seize-v.700222

improve performance of word sense disambiguation. They evaluated their system on Senseval-1 data.

3.1.6 Senseval-2 English Lexical Sample Task

The Senseval-2 [20] exercise was conducted in the summer of 2001. Systems were tested on eight languages besides English. The English Lexical Task for Senseval-2 consists of 4,328 instances for seventy three nouns, verbs and adjectives. Similar to Senseval-1 data, there exist a separate training and test corpus. Each instance is composed of one or two paragraphs and the target words are assigned senses from WordNet, version 1.7. The training corpus has 8,611 instances in all. Sample training and test instances from the Senseval-2 data is shown in Figures 33 and 34, respectively. The distributions of instances, as per the part of speech are depicted in Tables 26 for nouns, 27 for verbs and 28 for adjectives. The number of senses is from the training data. The data was validated to be in correct Senseval-2 data format by the `Sval2Check` [51] package. However, like *serve*, *interest* and Senseval-1 data, the program flagged 1 pair of instances (listed in Table 29) from the test data and 4 pairs of instances (listed in Table 30) from the training data to have the same context and sense ID (in case of training data) but differing instance IDs. This is a kind of duplication

Table 24: Senseval-1 training: same context and sense IDs but different instance IDs. Not removed.

Task	Instance 1	Instance 2	Sense
accident-n	accident-n.800754	accident-n.801240	532675
band-p	band-p.800971	band-p.800983	532745
excess-n	excess-n.800050	excess-n.800113	512404
float-v	float-v.800007	float-v.800095	523224
generous-a	generous-a.800009	generous-a.800270	512309
knee-n	knee-n.800025	knee-n.800174	516619
promise-a	promise-a.800536	promise-a.801100	537614
promise-n	promise-n.801208	promise-n.801758	537626
promise-n	promise-n.801345	promise-n.801766	537566
promise-n	promise-n.800723	promise-n.802037	538411
promise-v	promise-v.801094	promise-v.801496	537527
seize-v	seize-v.800113	seize-v.800298	507297
shake-p	shake-p.800481	shake-p.800977	504584
shirt-n	shirt-n.800244	shirt-n.800283	506479

Table 25: Senseval-1 training: same context but different sense IDs and instance IDs. Not removed.

INSTANCE 1			**INSTANCE 2**		
Task	**Instance**	**Sense**	**Task**	**Instance**	**Sense**
bet-v	bet-v.800135	519907	bet-n	bet-n.800135	519925
brilliant-a	brilliant-a.800178	brilliant-a.999997	brilliant-n	brilliant-n.800178	brilliant-n.999997
float-v	float-v.800068	523221	floating-a	floating-a.800068	523373
float-v	float-v.800008	523310	floating-a	floating-a.800008	523373
float-v	float-v.800181	523310	floating-a	floating-a.800181	523419
giant-a	giant-a.800294	giant-a.999997	giant-n	giant-n.800294	giant-n.999997
invade-v	invade-v.800044	invade-v.999997	invade-a	invade-a.800044	invade-a.999997
promise-a	promise-a.801790	promise-a.999997	promise-v	promise-v.801790	promise-v.999997
promise-a	promise-a.801790	promise-a.999997	promise-n	promise-n.801790	promise-n.999997
promise-n	promise-n.801790	promise-n.999997	promise-v	promise-v.801790	promise-v.999997
promise-a	promise-a.800340	promise-a.999997	promise-v	promise-v.800340	promise-v.999997
promise-a	promise-a.800340	promise-a.999997	promise-n	promise-n.800340	promise-n.999997
promise-n	promise-n.800340	promise-n.999997	promise-v	promise-v.800340	promise-v.999997
promise-a	promise-a.801997	promise-a.999997	promise-v	promise-v.801997	promise-v.999997
promise-a	promise-a.801997	promise-a.999997	promise-n	promise-n.801997	promise-n.999997
promise-n	promise-n.801997	promise-n.999997	promise-v	promise-v.801997	promise-v.999997
seize-v	seize-v.800213	seize-v.999997	seize-a	seize-a.800213	seize-a.999997
sack-v	sack-v.800188	sack-v.999997	sack-n	sack-n.800188	sack-n.999997
sack-v	sack-v.800108	sack-v.999997	sack-n	sack-n.800108	sack-n.999997

as described earlier. Due to the small number, these instances have been left as is. No other duplicates were found.

```
<lexelt item=”art.n”>
<instance id=”art.40001” docsrc=”bnc_ACN_245”>
<answer instance=”art.40001” senseid=”art<context>
Their multiscreen projections of slides and film loops have featured in orbital
parties, at the Astoria and Heaven, in Rifat Ozbek’s 1988/89 fashion shows, and
at Energy’s recent Docklands all-dayer.
From their residency at the Fridge during the first summer of love, Halo used
slide and film projectors to throw up a collage of op-art patterns, film loops of
dancers like E-Boy and Wumni, and unique fractals derived from video feedback.
&bquo;We’re not aware of creating a visual identify for the house scene, because
we’re right in there.
We see a dancer at a rave, film him later that week, and project him at the next
rave.&equo;
Ben Lewis Halo can be contacted on 071 738 3248.
<head>Art</head>you can dance to from the creative group called Halo
</context>
</instance>
</lexelt>
```

Figure 33: A sample training instance from Senseval-2 data.

Klein, Toutanova and Ilhan [30] developed a system for word sense disambiguation using and ensemble of heterogeneous classifiers. They evaluate their system using Senseval-2 data. The system achieved an accuracy of 61.7% in the Senseval-2 event. Along with Senseval-1 data, Senseval-2 data was used by Lee and Ng [33] in their experiments with a number of knowledge sources and supervised learning algorithms. Yarowsky and Florian [76] used the Senseval-2 data to compare sense disambiguation using six supervised learning algorithms and variations of the data representation. Pedersen [57] studies the the disambiguation of Senseval-2 data using ensemble decision trees. Mohammad and Pedersen [42] study the affect of guaranteed

<lexelt item=”art.n”>
<instance id=”art.40010” docsrc=”bnc_AHA_533”>
<context>
It is fair to say that nothing Frank has achieved as a film-maker approaches the
heights he scaled with The Americans. [/p] [p]
The 50-minute film he made for Arena suggests why: Last Supper — Frank on
Frank looks like a parody of the excesses of Sixties avant-garde film-making.
On an empty lot between two Harlem streets, a group of people arrive for an
outdoor party to celebrate the new publication of an unnamed author.
Initialy they make respectful small-talk about him, but their comments grow
increasingly resentful as it becomes apparent he will not show up. [/p] [p]
Some events in Last Supper appear roughly improvised.
Yet hefty chunks of dialogue are obviously — and rather archly — staged.
People say things like: &bquo;You ever notice how all <head>art</head>
focuses on people in trouble?&equo;
</context>
</instance>
</lexelt>

Figure 34: A sample test instance from Senseval-2 data.

Table 26: Instance distribution of Nouns in Senseval-2 data.

Word	Count		No. of Senses	Word	Count		No. of Senses
	Test	Train			Test	Train	
art	98	196	19	grip	51	102	7
authority	92	184	11	hearth	32	64	5
bar	151	304	22	holiday	31	62	8
bum	45	92	6	lady	53	105	10
chair	69	138	8	material	69	140	17
channel	73	145	10	mouth	60	119	12
child	64	129	9	nation	37	75	5
church	64	128	7	nature	46	92	9
circuit	85	170	16	post	79	157	15
day	145	289	18	restraint	45	91	9
detention	32	63	6	sense	53	107	9
dyke	28	58	4	spade	33	65	8
facility	58	114	6	stress	39	79	7
fatigue	43	85	8	yew	28	57	4
feeling	51	102	5				
TOTAL					**1754**	**3512**	**280**

Table 27: Instance distribution of Verbs in Senseval-2 data.

Word	Count		No. of Senses	Word	Count		No. of Senses
	Test	**Train**			**Test**	**Train**	
begin	280	557	8	match	42	86	8
call	66	132	23	play	66	129	25
carry	66	132	27	pull	60	122	33
collaborate	30	57	2	replace	45	86	4
develop	69	133	15	see	69	131	21
draw	41	82	32	serve	51	100	12
dress	59	119	14	strike	54	104	26
drift	32	63	9	train	63	125	9
drive	42	84	15	treat	44	88	6
face	93	186	7	turn	67	131	43
ferret	1	2	1	use	76	147	7
find	68	132	17	wander	50	100	4
keep	67	133	27	wash	12	25	13
leave	66	132	14	work	60	119	21
live	67	129	10				
TOTAL					**1806**	**3566**	**453**

Table 28: Instance distribution of Adjectives in Senseval-2 data.

Word	**Count**		**No. of Senses**
	Test	**Train**	
blind	55	108	9
colourless	35	68	3
cool	52	106	8
faithful	23	47	3
fine	70	142	13
fit	29	57	4
free	82	165	19
graceful	29	56	2
green	94	190	19
local	38	75	3
natural	103	206	25
oblique	29	57	14
simple	66	130	6
solemn	25	52	2
vital	38	74	8
TOTAL	**768**	**1533**	**138**

Table 29: Senseval-2 test data instances with the same context and sense IDs but different instance IDs. Not removed.

Task	**Instance 1**	**Instance 2**
collaborate	collaborate.063	collaborate.077

Table 30: Senseval-2 training: same context and sense IDs but different instance IDs. Not removed.

Task	Instance 1	Instance 2	Sense
collaborate.v	collaborate.002	collaborate.005	collaborate%2:41:00::
collaborate.v	collaborate.001	collaborate.036	collaborate%2:41:01::
dress.v	dress.001	dress.027	dress%2:29:01::
wander.v	wander.000	wander.044	wander%2:38:02::

part of speech pre-tagging of the head words of Senseval-2 data on the part of speech tags assigned to surrounding words by the Brill Tagger.

3.2 Pre-processing of Data for the Brill Tagger

Data in Senseval-2 format is part of speech tagged using the Brill Tagger [8] [9] [10]. The tagger requires the data to adhere to certain requirements in order to achieve accurate tagging. These requirements are listed below.

Tagger Req-I: All tokens constituting the data must be either words, numbers, punctuations or braces. The data must not contain tokens such as XML and SGML tags.

Tagger Req-II: One line per sentence.

Tagger Req-III: There should be no new line character within a sentence.

Tagger Req-IV: Apostrophe is to be tokenized as shown: Einstein's -> Einstein 's

Additionally, we believe that the quality of tagging will improve, if a few words are annotated with their correct part of speech. These words will be referred to as pre-tagged words. Intuitively, given that a word is tagged with the correct part of speech, there is a higher probability that the surrounding words are correctly tagged, since, the tag of a word is to some extent dependent on its neighbors. The part of speech tagged data is fed to the Collins parser [16] [37] to obtain parsed sentences. The parser places its own requirements on the data to be parsed, one of them being that the number of tokens per sentence must not exceed 120.

The *line, hard, serve, interest* and Senseval-1 data in the Senseval-2 data format and the original Senseval-2 data do not adhere to one or more of these requirements. Senseval-1 and Senseval-2 data have sentences which are on two or more lines. They also have instances with multiple sentences on the same line. The *line, hard, serve* and *interest* data have the sentence boundaries demarcated with <s> and </s> tags. The *line, hard, serve, interest*, Senseval-1 and Senseval-2 data, each consist of a few sentences which have more than 120 tokens. Additionally, the indeterminates in Senseval-1 data apart, the broad part of speech of the target words in all these instances is known. The `refine` [49] package has been developed to process any data in Senseval-2 data format in order to make it suitable for tagging by the Brill Tagger and parsing by the Collins parser. It may also be used to pre-tag the target words with appropriate part of speech tags. Its functions are listed below. Details follow:

I Restore split sentences

II Eliminate the <s> and </s> sentence boundary markers and place sentences on new lines.

III Detect multiple sentences on the same line and place them on new lines.

IV Pre-tag head words based on surface form.

V Supersede the surface form pre-tag with pre-tags suggested by the user for specific instances.

VI Replace contexts of user specified instance IDs with user specified instances.

3.2.1 Sentence Boundary

A pre-requisite in using the Brill Tagger is that there must be one sentence per line. The Senseval-1 and Senseval-2 data do not adhere to this requirement completely. `refine` [49] may be used to concatenate the split sentences. The lines to be concatenated are specified by their line numbers in the source text file. The *lines* files containing this information for Senseval-1 and Senseval-2 training and evaluation data has been created by manual inspection.

Split sentences apart, Senseval-1 and Senseval-2 data have certain instances with multiple sentences on the same line. This too does not conform to requirement II of the Brill Tagger. A simple sentence boundary detecting program, which is a part of `refine` [49] is used to generate copies of the training and evaluation data with one sentence per line.

The *line, hard, serve* and *interest* data in the Senseval-2 data format do not have instances with split sentences or improper sentence boundary demarcation. However, they do not all have one sentence per line. The sentences are separated from each other by sentence boundary markers *<s> and </s>* tags. Since, this does not conform with requirement 2 of the Brill Tagger, *refine* [49] is used to eliminate sentence boundary markers and place one sentence per line.

3.2.2 Pre-Tagging

The Brill Tagger is used to part of speech tag the data. The tagger works in two phases - initial state tagger and final state tagger. The initial state tagger assigns each word its most likely tag based on information it finds in a LEXICON. The final state tagger may transform this tag into another based on a set of contextual rules. A useful facility provided by the Brill Tagger is *pre–tagging*, which is the act of assigning parts of speech to tokens in a text before tagging the complete text with a tagger. We shall call the part of speech of the word being pre-tagged, the pre–tag. Since, we are provided with the part of speech of the head words in the data we use, this provides an effective mechanism to correctly tag the head words and use these tags to better tag the tokens around. The latter gains prominence by the nature of tags which are dependent, to a certain extent, on the tags of the surrounding tokens. All target words in *line* and *interest* data are nouns. Similarly, *hard* data target words are adjectives and *serve* data target words are verbs. The part of speech of the Senseval-1 and Senseval-2 data instances is also known. `refine` [49] has been used to make use of this information and appropriately pre-tag the head words of *line, hard, serve, interest*, Senseval-1 and Senseval-2 data. The pre-tagged data created by `refine` [49] is in a format acceptable by the Brill Tagger. The pre-tagged sample instance from *line* data is shown in Figure 35. The target word *lines* is pre-tagged to be a plural noun (NNS).

The pre-tagging may be done at two levels. Firstly, all head words are pre-tagged based on their surface form. For example, the various surface forms of *eat* for example are *eat, ate, eaten* and *eats* and occurrences

<instance id=”line-n.w7_010:888:”>
<answer instance=”line-n.w7_010:888:” senseid=”cord”/>
<context>
<s> The company argued that its foreman needn’t have told the worker not to move the plank to which his lifeline was tied because ”that comes with common sense. ” </s> <@> <@> <p> <@> <s> The commission noted, however, that Dellovade hadn’t instructed its employees on how to secure their lifelines and didn’t heed a federal inspector’s earlier suggestion that the company install special safety <head>lines//**NNS** </head> inside the A-frame structure it was building. </s> </context>
</instance>

Figure 35: Pre-tagged *line* data instance in Senseval-2 data format.

in different surface forms may be pre-tagged to different parts of speech. The package uses a file with a list of the surface forms and their most likely part of speech. The file is analogous to the LEXICON file of the Brill Tagger. Each token within the head tags is considered for pre-tagging. If there exists an entry for the head word, it is pre-tagged with the associated most likely part of speech. It may be noted that head words with apostrophe are first tokenized as shown in Table 31 before pre-tagging. This is in accordance with requirement 4 of the Brill Tagger.

Table 31: Pre-tagging of head words with apostrophe

	Head word fragment
Original fragment	<head>band’s</head>
Tokenization	<head>band ’s</head>
Pre-tagging	<head>band//NN ’s</head>

The surface form based pre-tagging files have been created manually for the *line, hard, serve, interest*, Senseval-1 and Senseval-2 data. It may be noted that the broad part of speech of the head words in the *line, hard, serve, interest*, Senseval-1 and Senseval-2 data are known. For example, *line* is a noun, *serve* is a verb and natural - from the Senseval-2 data - is an adjective. The exact part of speech corresponding to various surface forms has been chosen considering the broad part of speech. In case of nouns, the surface form which is the same as the root word is tagged as a *common noun* (NN). For example, the surface form *authority* matches the root form of the word, which is *authority* as well. Hence, head words whose surface form is *authority* are tagged to be a *common noun*. Surface forms which correspond to the plural form of the word, for example *authorities*, are tagged as *plural nouns* (NNS). Surface forms which are capitalized are tagged as *proper noun* (NNP) or *plural proper noun* (NNPS) depending on whether it matches the root word or corresponds to the plural form of the word. Examples would be *Authority* as in *National Aviation Authority* and *Authorities* as in *American Association of Port Authorities*. Verbs have been pre-tagged into five parts of speech. Verbs which match the root form are tagged as `base form verb` (VB), for example *take*. Surface forms which correspond to the past tense such as *took* are pre-tagged as `verb past` (VBD). Surface forms which end in *ing* and correspond to the gerund are tagged *gerund* (VBG), for example *taking*. The *past participle* (VBN) tag is assigned to surface forms such as *taken*. Surface forms which are used when referring to a third person such as *takes* are tagged `verb present, 3rd person` (VBZ). The adjectives are tagged as *adjectives* (JJ). It may be noted that there exists verbs with the same past tense form and past participle form, for example, consider the verb *call*, both past and past participle form have the same surface form *called*. This may be a cause of error but the authors believe this is a small price to pay for the benefits of pre-tagging. Without pre-tagging head words which we know belong to a certain broad part of speech may be tagged into a totally different class altogether. Pre-tagging eliminates this problem.

All head word instances with the same surface form are tagged alike by the surface form based pre-tagging. Besides the special case of verbs mentioned above, this may cause errors in case of nouns as well. Proper nouns are identified based on capitalization, but words which are not proper nouns are capitalized when they are at the start of the sentence. Thus, in the case of capitalized nouns which are at the start of a sentence, there is no way of determining the correct pre-tag without manually examining the sentence. The second level of pre-tagging involves overriding the surface form based pre-tag of specific instances by user specified pre-tag. The pre-tag to be assigned to the instance and line number of the head word in the data file are specified in the *specific pre-tag file*. The context of all capitalized noun head words in the Senseval-1 and Senseval-2 data has been manually examined to determine the most suitable part of speech of the head word. The *specific pre-tag file* for these instances has been created and provided with the package in order to facilitate duplication of the pre-tagging.

`refine` [49], as mentioned above, has many functions which make the data more suitable for part of speech tagging by the Brill Tagger. However, the processed files are still in Senseval-2 data format. The next section describes the pre-processing of data in Senseval-2 data format to produce text files acceptable by the Brill Tagger.

3.2.3 Senseval-2 Format to the Format Acceptable by Brill Tagger

Data in Senseval-2 format has numerous XML and SGML tags. These tags are not acceptable inputs to the Brill Tagger. If left as is, the tagger will assign part of speech tags to them as well, instead of ignoring them. These tags will then affect the selection of tags of surrounding words, which is undesirable. As part of this thesis we developed the *posSenseval* [48] package which may be used to part of speech tag any data in Senseval-2 data format. It has a pre-processing stage which involves the following steps:

I Removal of XML, SGML tags and other non-contextual information such as representation of vocal sounds and typographical errors.

II Conversion of character references to appropriate symbols - *<* to <

III Removal of all other character references.

IV Appropriate tokenization. Special consideration for *apostrophe* - *Einstein's* to *Einstein 's*. Hyphenated words like *passers-by* and *avant-garde*, are not to be split up.

In order to achieve the best results, the Brill Tagger must be given only contextual tokens. Thus, all XML and SGML tags, which are not contextual in nature, are eliminated. The XML tags are identified by the presence of angular brackets around them. For example <head>. The SGML tags are identified as tokens with square braces around them. For example *[/quote]*. Character references are a way to refer to a character which is independent of the way the actual character is entered. For example: *[* refers to the left square bracket [and *]* to [. Such references are useful to differentiate between different usages of the same character. The symbol '[' for example may be used as part of the sentence or as part of an SGML tag. As per the Senseval-2 data format, character references are used to refer to those symbols which may be used as part of XML or SGML tags. '<' and '>' are the other two symbols which must be entered as character references, if part of a sentence. ‘<’ and ‘>’ are their corresponding character references, respectively.

Senseval-1 data has square braces which should have ideally been specified by their corresponding character references. They are converted to corresponding character references in the pre-processing step - [to *[* and] to *]*. These exceptions are identified by the presence of multiple tokens within the square braces and the conversion to appropriate character references is done. Senseval-1 data has certain non-contextual information such as representations of vocal sounds and typographical errors within curly braces. For example {*vocal sound=”um”*} and {*typo bad=”amazes”,good=”<head>amazed</head>”*}. All such tokens are eliminated before tagging. In case of typographical errors, the correctly spelled token is put in place of the deleted tokens. As with the square braces, if the curved braces are a part of the context, they are to be specified by their appropriate character references - *{* and *}* represent left and right curly braces, respectively.

The *LEXICON* provided with the Brill Tagger has a list of tokens and their most likely part of speech tag. The other possible parts of speech in which the word might occur are also specified. Table 32 lists some example entries. During the first phase of tagging, the tagger assigns each word the most frequent tag associated with it as picked up from this file. Words which do not have entries in the LEXICON are assigned proper nouns. The LEXICON does not have entries for character references. Thus, the tagger has no information regarding its most likely tag. It does however, have entries for symbols corresponding to symbols which correspond to certain character references. These entries are depicted in Table 33. It may be noted that even though certain symbols need not be specified as character references, they may still be done so in the data. Such character references if found in the text are converted back to their appropriate symbols. All other character references are eliminated. It may be noted that the LEXICON provided by the Brill Tagger has been automatically derived and hence has irregular coverage.

Yet another important step of the pre-processing is tokenization. The Brill Tagger requires all words and punctuations to be space separated - requirement 4. The only exception being an apostrophe which is tokenized differently. The desired tokenization is exemplified below.

Before Tokenization: *Einstein***'s** *mass and energy formula was revolutionary***;** *or was it relativity***?**
After Tokenization: *Einstein* **'s** *mass and energy formula was revolutionary* **;** *or was it relativity* **?**

Table 32: Example entries in the LEXICON

Type	Most Frequent Tag	Other Possible Tags
brown	JJ	NN VB
meeting	NN	VBG
meetings	NNS	–

Table 33: Entries in the LEXICON corresponding to certain character references.

	LEXICON Entry		
Character Reference	**Type**	**Most Frequent Tag**	**Other Possible Tags**
&	&	CC	NNP SYM
'	’	POS	NN NNPS ”
{	{	(	
}	}	)	
[	[	(	
]	]	)	
‘	‘	“	
’	’	POS	NN NNPS ”
<	<	SYM	
>	>	SYM	NN
–	-	:	
#	#	#	

It may be noted that as part of tokenization, hyphenated words like *passers-by* and and *avant-garde*, are not split up. This is because, by hyphenation these words take on a special meaning and are to be treated as one token. The text file created after the pre-processing conforms to the requirements of the Brill Tagger ensuring reliable part of speech tagging. The example instance in Senseval-2 data format from Figure 33 after pre-processing is shown in Figure 36. This sentence is fed directly to the Brill Tagger.

Their multiscreen projections of slides and film loops have featured in orbital
parties , at the Astoria and Heaven , in Rifat Ozbek 's 1988 / 89 fashion shows
, and at **Energy 's** recent Docklands **all-dayer** .
From their residency at the Fridge during the first summer of love , Halo used
slide and film projectors to throw up a collage of op-art patterns , film loops
of dancers like **E-Boy** and Wumni , and unique fractals derived from video
feedback .
" **We 're** not aware of creating a visual identify for the house scene ,
because **we 're** right in there .
We see a dancer at a rave , film him later that week , and project him at the
next rave . "
Ben Lewis Halo can be contacted on 071 738 3248 .
Art//NNP you can dance to from the creative group called Halo

Figure 36: Senseval-2 instance from Figure 33 after being pre-processed by `posSenseval` making it suitable for the Brill Tagger.

3.3 Part of Speech Tagging

The pre-processed data described in the previous section is given to the Brill Tagger which part of speech tags it. The part of speech tagged instance from Figure 36 is shown in Figure 37.

Their/PRP$ multiscreen/JJ projections/NNS of/IN slides/NNS and/CC film/NN loops/NNS have/VBP featured/VBN in/IN orbital/JJ parties/NNS ,/, at/IN the/DT Astoria/NNP and/CC Heaven/NNP ,/, in/IN Rifat/NNP Ozbek/NNP ’s/POS 1988/CD //NN 89/CD fashion/NN shows/VBZ ,/, and/CC at/IN Energy/NNP ’s/POS recent/JJ Docklands/NNS all-dayer/JJR ./.
From/IN their/PRP$ residency/NN at/IN the/DT Fridge/NNP during/IN the/DT first/JJ summer/NN of/IN love/NN ,/, Halo/NNP used/VBD slide/NN and/CC film/NN projectors/NNS to/TO throw/VB up/IN a/DT collage/NN of/IN op-art/JJ patterns/NNS ,/, film/NN loops/NNS of/IN dancers/NNS like/IN E-Boy/NNP and/CC Wumni/NNP ,/, and/CC unique/JJ fractals/NNS derived/VBN from/IN video/NN feedback/NN ./.
”/” We/PRP ’re/VBP not/RB aware/JJ of/IN creating/VBG a/DT visual/JJ identify/VB for/IN the/DT house/NN scene/NN ,/, because/IN we/PRP ’re/VBP right/NN in/IN there/RB ./.
We/PRP see/VBP a/DT dancer/NN at/IN a/DT rave/VBP ,/, film/NN him/PRP later/RB that/DT week/NN ,/, and/CC project/NN him/PRP at/IN the/DT next/JJ rave/VBP ./. ”/” Ben/NNP Lewis/NNP Halo/NNP can/MD be/VB contacted/VBN on/IN 071/CD 738/CD 3248/CD ./.
Art//NNP you/PRP can/MD dance/VB to/TO from/IN the/DT creative/JJ group/NN called/VBN Halo/NNP

Figure 37: Part of speech tagged instance from Figure 36

3.4 Part of Speech Tagging - Post Processing

Once, the data has been part of speech tagged by the tagger, the pos-processing phase involves three steps:

I: Converting the tokens which have been part of speech tagged to lower case.

II: Putting back the XML and SGML tokens, which were removed during pre-processing.

III: Putting back the character references eliminated during pre-processing.

IV: Placing the part of speech tags in angular braces.

V: Checking if any of the pre-tags assigned to the head words, have been overridden to other tags by the tagger.

3.4.1 Capitalization

Capitalization is an important cue, which the Brill Tagger uses to correctly part of speech tag the data. Capitalized nouns are more likely to be proper nouns than not. However, many a natural language tools like word sense disambiguation systems, might want to treat the different capitalizations as the same token. For example if the *industrial plant* helps identify the correct sense of *plant*, so does *Industrial plant* or *INDUSTRIAL PLANT*. Thus, as part of the post processing, all tokens which were part of speech tagged are converted to lower case.

3.4.2 XML'izing

As a matter of principle, `posSenseval` [48] does not permanently delete any of the tokens in the data file. All the tokens, XML and SGML tags, which are eliminated as part of the pre-processing step are put back into the data file at corresponding positions. The XML tags like the instance ID tag, context tag and head word tags are basic components of the Senseval-2 data format and have to be placed back exactly as they were in order for the final file to conform to the data format. The latter is desired since, our disambiguation system apart, any other system which accepts data in Senseval-2 data format may use the part of speech tagged data created. The SGML tags are not being used by our disambiguation system.

These tags are bracketed with angular braces so as to XML'ize them, before putting them back. For example *[caption]* is put back as <*[caption]*>. This is useful since programs which would like to skip XML and SGML tokens may do so, simply, by ignoring what is inside the angular braces. The representations of vocal sounds and typographical errors which have been encoded in Senseval-1 data in curly braces, are also put back after bracketing them in angular braces. For example, {*vocal sound=”um”*} i sput back as <{*vocal sound=”um”*}>. The character references whose corresponding symbols did not have entries in Brill Tagger's LEXICON and which were removed during pre-processing are also put back after bracketing with angular braces. For example, *þ* is put back as <*þ*>.

As seen in Figure 37, the Brill Tagger attaches the part of speech to each token with a forward slash. Since, the part of speech tags, like the XML and SGML tokens gives information about the tokens and are not part of the sentence, they are XML'ized as well. The sample instance from Figure 37 after putting back the post-processing is depicted in Figure 38. Note: Only a part of the instance is shown due to space constraints. The *p* immediately following the ‘<’ symbol suggests that this XML tag has the part of speech of the previous token.

3.4.3 Examining the Pre-tags

The head words were pre-tagged with their parts of speech before giving the data to the Brill Tagger. Casual examination of the tagged data revealed that some of the pre-tags were changed to other parts of speech and in many cases a change not just within the broad part of speech such as VBD to VBZ but more radical as in across the broad parts of speech from verb to noun. We shall refer to such erroneous tags as mis-tags and the errors as radical errors when the change is across a broad part of speech and subtle errors when the change is within a broad part of speech. For example from VBD to NN and VBD to VBZ, respectively. Our word sense disambiguation system uses the parts of speech of words surrounding the target word. An error in the part of speech of the target word will likely cause an error in the part of speech of the surrounding words, affecting sense disambiguation. Additionally, other tools which base their results on the parts of speech will be affected as well (in our case the Collins parser). The gravity of the matter lead to a thorough examination of the number of such mis-taggings. Table 34 lists the number of such errors and the percentage of errors with respect to the total number of target words.

Since, the quality of the part of speech tagging, especially of the head word and those surrounding it is vital,

<lexelt item=”art.n”>
<instance id=”art.40001” docsrc=”bnc_ACN_245”>
<answer instance=”art.40001” senseid=”art<context>
” <p=”””/> we <p=”PRP”/> ’re <p=”VBP”/> not <p=”RB”/> aware <p=”JJ”/> of <p=”IN”/>
creating <p=”VBG”/> a <p=”DT”/> visual <p=”JJ”/> identify <p=”VB”/> for <p=”IN”/> the
<p=”DT”/> house <p=”NN”/> scene <p=”NN”/> , <p=”,”/> because <p=”IN”/> we <p=”PRP”/>
’re <p=”VBP”/> right <p=”NN”/> in <p=”IN”/> there <p=”RB”/> . <p=”.”/>
we <p=”PRP”/> see <p=”VBP”/> a <p=”DT”/> dancer <p=”NN”/> at <p=”IN”/> a <p=”DT”/> rave
<p=”VBP”/> , <p=”,”/> film <p=”NN”/> him <p=”PRP”/> later <p=”RB”/> that <p=”DT”/> week
<p=”NN”/> , <p=”,”/> and <p=”CC”/> project <p=”NN”/> him <p=”PRP”/> at <p=”IN”/> the
<p=”DT”/> next <p=”JJ”/> rave <p=”VBP”/> . <p=”.”/> ” <p=”””/>
<[hi]> ben <p=”NNP”/> lewis <p=”NNP”/> <[/hi]> halo <p=”NNP”/> can <p=”MD”/> be
<p=”VB”/> contacted <p=”VBN”/> on <p=”IN”/> 071 <p=”CD”/> 738 <p=”CD”/> 3248
<p=”CD”/> . <p=”.”/>
<[ptr]><[/p]> <[caption]> <head>art <p=”NNP”/></head> you <p=”PRP”/> can <p=”MD”/>
dance <p=”VB”/> to <p=”TO”/> from <p=”IN”/> the <p=”DT”/> creative <p=”JJ”/> group
<p=”NN”/> called <p=”VBN”/> halo <p=”NNP”/> <[/caption]> <[/div2]> <[div2]> <[head]>
</context>
</instance>
</lexelt>

Figure 38: The instance from Figure 37 after post-processing.

Table 34: Radical and Subtle errors in the part of speech tags of the head words.

Sense-Tagged Data	**Target Words**	**Radical Errors**	**Percentage**	**Subtle Errors**	**Percentage**
Senseval-2 Training	8611	347	4.0%	746	8.7%
Senseval-2 Test	4328	185	4.3%	398	9.2%
Senseval-1 Training	13276	353	2.7%	1165	8.8%
Senseval-1 Test	8452	209	2.5%	821	9.7%
line	4149	85	2.0%	278	6.7%
hard	4337	145	3.3%	0	0.0%
serve	4378	89	2.0%	815	18.6%
interest	2476	74	3.0%	0	0.0%
TOTAL	**50,007**	**1487**	**3.0%**	**4223**	**8.4%**

we developed a patch BrillPatch [41] to the Brill Tagger which guarantees that the pre-tag be respected all through the tagging process and contribute to the selection of appropriate tags of the surrounding words. On using guaranteed pre-tagging, as expected, we found no mistags in all the data. Also, we expect the quality of parsing and word sense disambiguation to improve.

3.5 Collins Parser

The part of speech tagged data is fed to the Collins parser [16] [37] to obtain parsed sentences. The Collins parser places its own requirements on the data to be parsed which are as follows:

Parser Req-I: Number of tokens per sentence must not be more than 120.

Parser Req-II: There must be one sentence per line.

Parser Req-III: All tokens must be followed by white space and corresponding part of speech.

Parser Req-IV: Total number of tokens in the sentence must be placed at start of sentence.

Parser Req-V: The data file should not have more than 2500 sentences.

Requirement I, as described earlier, is taken care by `refine` [49]. We developed the `parseSenseval` [47] package to pre-process the data so that it to parse the data. It accepts part of speech tagged sentences in a format as output by the Brill Tagger. It has the following functions:

I Pre-process the data to make it acceptable by the Collins parser.

II Use the Collins parser to parse the data.

III XML'ize the parsed output.

IV Given the source file in Senseval-2 data format, put back the XML and SGML tags.

3.5.1 Preprocessing for the Collins Parser

Some of the sentences in Senseval-1, Senseval-2, *line, hard, serve* and *interest* data have very long sentences (more than 120 tokens) as detailed in Table 35. The Collins parser which is used to parse the data does not accept sentences which have more than 120 tokens. Hence another option was incorporated into `refine` [49] to allow replacements of contexts of user specified instances with modified/new contexts. The long sentences were split into two (possibly more) sentences by manual inspection. Contexts having these long sentences were replaced by contexts having manually split sentences. Manual splits were made at colons, semicolons, quotation marks, hyphens or conjunctions.

The instances whose contexts are to be replaced and the modified/new contexts are specified via a CONTEXT file. The CONTEXT file must have the instance ID of the instance whose context is to be replaced followed by the modified/new context. The context must be demarcated by <context> and </context> tags. The instance ID, <context> and <context> tags must be on new lines. Blank lines are allowed, however, everything between the context tags is considered part of context. The token(s) between the <head> and </head> tags of the context in the CONTEXT file are replaced by corresponding tokens in the SOURCE file. If the original context has its head word pre-tagged with a part of speech, the updated context will thus have the pre-tag. CONTEXT files corresponding to the Senseval-1, Senseval-2 test and training data, *line, hard, serve* and *interest* data were created.

Table 35: Instances with long sentences (more than 120 tokens).

Sense-Tagged Data	**Total number of Instances**	**Instances with Long Sentences**	**Percentage**	**Head word part of Long Sentence**	**Percentage**
Senseval-2 Training	8611	27	0.31%	16	0.18%
Senseval-2 Test	4328	18	0.42%	9	0.21%
Senseval-1 Training	13276	54	0.41%	43	0.32%
Senseval-1 Test	8452	43	0.51%	36	0.43%
line	4149	1	0.02%	1	0.02%
hard	4337	2	0.05%	2	0.05%
serve	4378	3	0.07%	1	0.02%
interest	2476	1	0.04%	1	0.04%
TOTAL	**50,007**	**149**	**0.30%**	**109**	**0.22%**

Preprocessing of the part of speech tagged data as given out by the Brill Tagger is done to bring it in a format acceptable by the Collins parser. The number of tokens in each sentence is counted and placed at the start of the sentence. The forward slash separating the token and the corresponding part of speech tag is replaced by a space. Additionally, certain part of speech tags as given out by the Brill Tagger are not known to the Collins parser, hence, these tags are replaced by their closest related part of speech tag. Details of these replacements are listed in Table 36.

Table 36: Brill Part of Speech Tags unknown to the Collins Parser and their replacements.

Brill Tagger POS Tag	**Replacement for Collins Parser**
(	SYM
)	SYM
/	CC
”	”

The two consecutive forward slashes indicating pre-tagging are replaced by a single forward slash. If the file is has more than 2400 lines it is broken into multiple files, each with 2400 lines, except the last. This is because the Collins Parser does not accept files with more than 2500 lines. each of the files is parsed independently. The instance from Figure 37 after pre-processing and ready to parse is shown in Figure 39.

3.5.2 Parsing with the Collins Parser

The pre-processed data is parsed using the Collins parser with its default option. INPUT signifies the input to the parser while OUTPUT is the parsed data file. PARSERHOME is where the parser is downloaded.

$$gunzip\ -c\ PARSERHOME/models/model3/events.gz\ \mid\ PARSERHOME/code/parser$$

$$./INPUT\ PARSERHOME/models/model3/grammar\ 10000\ 1\ 1\ 1\ 1\ >\ ./OUTPUT$$

The last sentence (has the head word) from the instance shown in Figure 39 is shown in Figure 40 after parsing.

38 Their PRP$ multiscreen JJ projections NNS of IN slides NNS and CC film NN loops NNS have VBP featured VBN in IN orbital JJ parties NNS , , at IN the DT Astoria NNP and CC Heaven NNP , , in IN Rifat NNP Ozbek NNP ’s POS 1988 CD / NN 89 CD fashion NN shows VBZ , , and CC at IN Energy NNP ’s POS recent JJ Docklands NNS all-dayer JJR . .
45 From IN their PRP$ residency NN at IN the DT Fridge NNP during IN the DT first JJ summer NN of IN love NN , , Halo NNP used VBD slide NN and CC film NN projectors NNS to TO throw VB up IN a DT collage NN of IN op-art JJ patterns NNS , , film NN loops NNS of IN dancers NNS like IN E-Boy NNP and CC Wumni NNP , , and CC unique JJ fractals NNS derived VBN from IN video NN feedback NN . .
22 “ “ We PRP ’re VBP not RB aware JJ of IN creating VBG a DT visual JJ identify VB for IN the DT house NN scene NN , , because IN we PRP ’re VBP right NN in IN there RB . .
23 We PRP see VBP a DT dancer NN at IN a DT rave VBP , , film NN him PRP later RB that DT week NN , , and CC project NN him PRP at IN the DT next JJ rave VBP . . “ “
11 Ben NNP Lewis NNP Halo NNP can MD be VB contacted VBN on IN 071 CD 738 CD 3248 CD . .
11 Art NNP you PRP can MD dance VB to TO from IN the DT creative JJ group NN called VBN Halo NNP

Figure 39: Instance from Figure 37 after pre-processing and ready to parse

```
PROB 3371 −76.5798 0

TOP −76.5798 S −70.1835 NP−A −51.1724 NPB −0.873273 NN 0 Art

                    SBAR−g −41.3336 Ss−A−g −41.2892 NP−A −0.0053634 NPB −0.00170351 PRP 0 you

                                        VP−g −38.0742 MD 0 can

                                               VP−A−g −27.9592 VB 0 dance

                                                      PP−g −0.686837 TO 0 to

                                               PP −17.7415 IN 0 from

                                                   NP−A −11.7417 NPB −11.2813 DT 0 the

                                                            JJ 0 creative

                                                            NN 0 group

          VP −6.1979 VBD 0 called

                S−A −1.79597 NP−A −1.16349 NPB −0.869919 NNP 0 Halo
```

```
(TOP~called~1~1 (S~called~2~2 (NP−A~Art~2~1 (NPB~Art~1~1 Art/NN ) (SBAR−g~can~1~1
(Ss−A−g~can~2~2 (NPB~you~1~1 you/PRP ) (VP−g~can~2~1 can/MD (VP−A−g~dance~3~1 dance/VB
(PP−g~to~2~1 to/TO T/TRACE ) (PP~from~2~1 from/IN (NPB~group~3~3 the/DT creative/JJ group/NN
) ) ) ) ) )) (VP~called~2~1 called/VBD (S−A~Halo~1~1 (NPB~Halo~1~1 Halo/NNP ) ) ) ) )
```

Figure 40: Sentence from Figure 39 after parsing by the Collins Parser

3.5.3 Post Processing - Beyond the Collins Parser

Similar to the post processing done by `posSenseval` [48], the post-processing in `parseSenseval` involves XML'izing the parse information. Only the parenthesized version of the parse is retained. All tokens which have the parse information have either the left or right bracket. All these tokens are placed in angular braces. The forward slash separating a word and its part of speech is replaced by a space and the part of speech tag is placed in angular braces as well. The word is converted to lower case. The sample sentence from Figure 40 after XML'izing is shown in Figure 41.

<P="TOP~called~1~1"> <P="S~called~2~2"> <P="NP-A~Art~2~1"> <P="NPB~Art~1~1"> Art <p="NN"/> </P> <P="SBAR-g~can~1~1"> <P= "Ss-A-g~can~2~2"> <P="NPB~you~1~1"> you <p="PRP"/> </P> <P="VP-g~can~2~1"> can <p="MD"/> <P="VP-A-g~dance~3~1"> dance <p="VB"/> <P="PP-g~to~2~1"> to <p="TO"/> </P> <P="PP~from~2~1"> from <p="IN"/> <P="NPB~group~3~3"> the <p="DT"/> creative <p="JJ"/> group <p="NN"/> </P> </P> </P> </P> </P> </P> </P> <P="VP~called~2~1"> called <p="VBD"/> <P="S-A~Halo~1~1"> <P="NPB~Halo~1~1"> Halo <p="NNP"/> </P> </P> </P> </P>

Figure 41: Sentence from Figure 40 after XML'izing.

If the original Senseval-2 format data file is provided, all the XML and SGML tags are put back at their original positions in the parsed file. The SGML tags, as in `posSenseval` [48] are placed in angular braces. If the original data file was split into multiple files, each with 2400 lines, the files are concatenated, back into one large file. The sample instance from Figure 40 after post-processing is shown in Figure 42. Only the sentence housing the head word is shown due to space constraints.

```
<lexelt item=”art.n”>
<instance id=”art.40001” docsrc=”bnc_ACN_245”>
<answer instance=”art.40001” senseid=”art<context>
...
...
<P=”TOP˜called˜1˜1”> <P=”S˜called˜2˜2”> <P=”NP-A˜Art˜2˜1”> <P=”NPB˜Art˜1˜1”>
<head> art </head> <p=”NN”/> </P> <P=”SBAR-g˜can˜1˜1”> <P=”Ss-A-g˜can˜2˜2”>
<P=”NPB˜you˜1˜1”> you <p=”PRP”/> </P> <P=”VP-g˜can˜2˜1”> can <p=”MD”/>
<P=”VP-A-g˜dance˜3˜1”> dance <p=”VB”/> <P=”PP-g˜to˜2˜1”> to <p=”TO”/> </P>
<P=”PP˜from˜2˜1”> from <p=”IN”/> <P=”NPB˜group˜3˜3”> the <p=”DT”/> creative
<p=”JJ”/> group <p=”NN”/> </P> </P> </P> </P> </P> </P> </P>
<P=”VP˜called˜2˜1”> called <p=”VBD”/> <P=”S-A˜Halo˜1˜1”> <P=”NPB˜Halo˜1˜1”>
halo <[/caption]> <[/div2]> <[div2]> <[head]> <p=”NNP”/> </P> </P> </P>
</context>
</instance>
</lexelt>
```

Figure 42: Sentence from Figure 40 after Post-Processing.

4 EXPERIMENTS

This section describes the Word Sense Disambiguation experiments based on lexical and syntactic features. The syntactic features may be categorized into part of speech features and parse features. The experiments have been carried out on part of speech tagged and parsed Senseval-2, Senseval-1, *line, hard, serve* and *interest* data. The part of speech tagging is done using `posSenseval` [48] with the Brill Tagger and Guaranteed Pre-Tagging [42] while the parsing is done using `parseSenseval` [47] with the Collins Parser. Senseval-1 and Senseval-2 have pre-determined test and training data sets which have been utilized in these experiments. The *line, hard, serve* and *interest* data do not have a pre-determined division into test and training data and are hence split up randomly into test and training sets using `setup.pl`, a part of the `SenseClusters` [60] package. The training set has 80% of the instances while the test set has the remaining 20%. The C4.5 algorithm as implemented by Waikato Environment for Knowledge Analysis *Weka*[23][72], is used in the thesis to learn a decision tree for each word to be disambiguated. The Weka implementation is in Java and is referred to as J48. Programs from SenseTools [5] are utilized to do the word sense disambiguation and evaluation using Weka.

4.1 Individual Features

4.1.1 Lexical Features

The lexical features we have studied are surface form of the target word, unigrams and bigrams. Pedersen [55] has shown that lexical features can attain very good accuracies and these experiments are a reproduction of those experiments on Senseval-1 and Senseval-2 data. Similar experiments on *line, hard, serve* and *interest* data have been conducted for the first time. Tables 37, 38 and 39 display the accuracies for Senseval-2, Senseval-1 and *line, hard, serve* and *interest* data, respectively. It may be noted that the system attempts at identifying the intended sense of all instances and hence the recall is 100%. The accuracy of the majority classifier, which always guesses the intended sense to be the majority sense in the training data, is also specified as a point of comparison. In case of Senseval-2 and Senseval-2 data, a break down of the accuracies for each part of speech is depicted as well.

As may be observed, the bigrams and unigrams perform well for all the data. The results for Senseval-1 and

Table 37: Accuracy using Lexical Features on Senseval-2 Data.

	All	**Nouns**	**Verbs**	**Adjectives**
Majority Classifier	47.7%	51.0%	39.7%	59.0%
Surface Form	49.3%	54.6%	40.1%	59.0%
Unigrams	55.3%	61.6%	46.8%	60.9%
Bigrams	55.1%	60.9%	48.6%	61.8%

Table 38: Accuracy using Lexical Features on Senseval-1 Data.

	All	**Nouns**	**Verbs**	**Adjectives**	**Indeterminates**
Majority Classifier	56.3%	57.2%	56.9%	64.3%	43.8%
Surface Form	62.9%	67.1%	59.8%	68.5%	57.2%
Unigrams	66.9%	72.3%	62.7%	70.2%	63.2%
Bigrams	66.9%	71.7%	65.6%	69.9%	59.4%

Table 39: Accuracy using Lexical Features on *line, hard, serve* and *interest* Data.

	line	**hard**	**serve**	**interest**
Majority Classifier	54.3%	81.5%	42.2%	54.9%
Surface Form	54.3%	81.5%	44.2%	64.0%
Unigrams	74.5%	83.4%	73.3%	75.7%
Bigrams	72.9%	89.5%	72.1%	79.9%

Senseval-2 are comparable those obtained by Pedersen [55]. Surface form which is a very simplistic feature does not do significantly better than the majority classifier, except in case of Senseval-1 data. The part of speech split up of results for Senseval-1 and Senseval-2 data demonstrates that the unigrams and bigrams perform far better for nouns and verbs as compared to adjectives. Figures 43, 44 and fig:bi-tree depict some of the decision trees learnt for surface form, unigrams and bigrams.

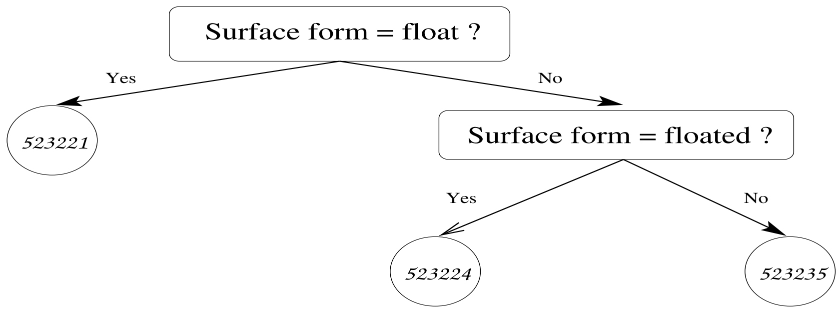

Figure 43: Sample Decision Tree learnt using Surface Form as features

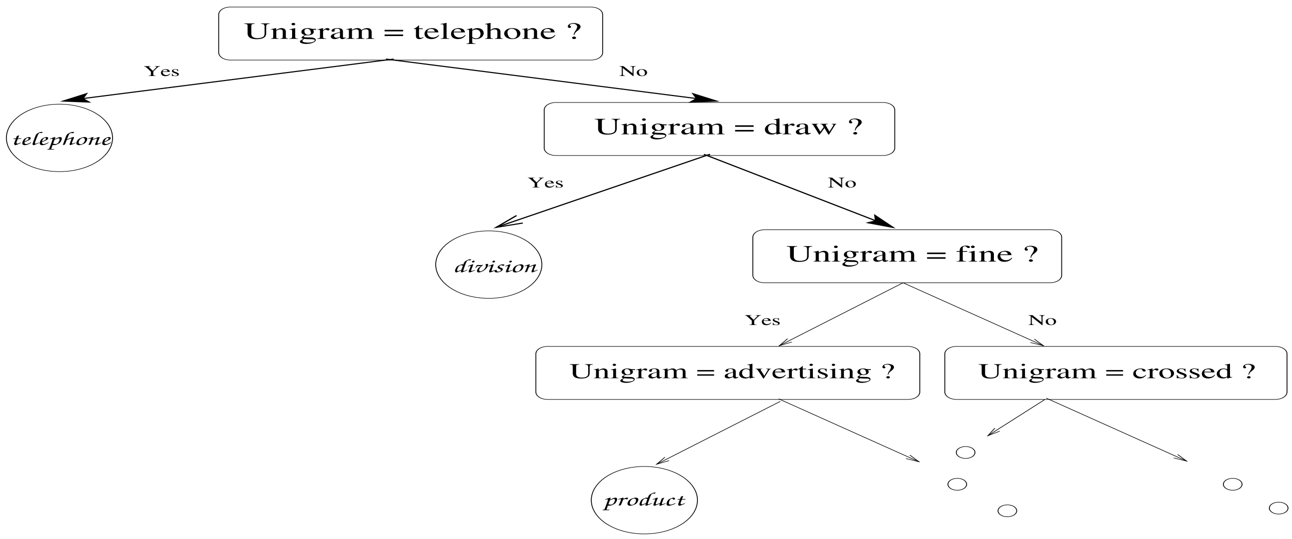

Figure 44: Sample Decision Tree learnt using Unigrams as features

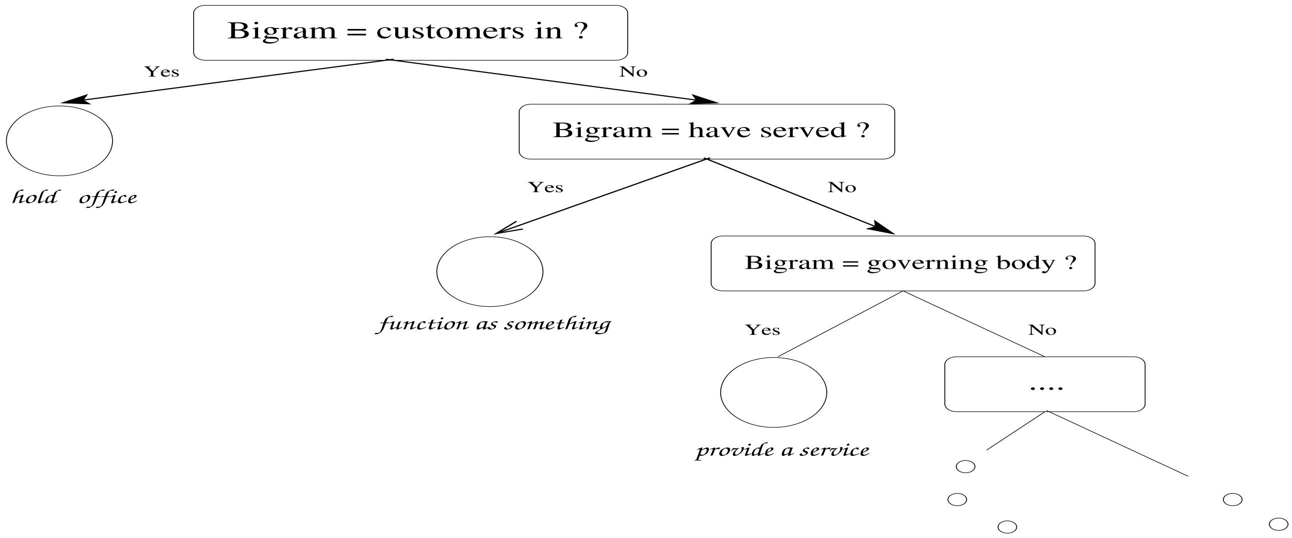

Figure 45: Sample Decision Tree learnt using Bigrams as features

4.1.2 Part of Speech Features

Individual part of speech incorporates the part of speech of one word at a particular position relative to the target word. Tables 40, 41, 42 show disambiguating accuracies using individual word parts of speech on Senseval-2, Senseval-1 and *line, hard, serve* and *interest* data, respectively. The system will always produce a sense for a test instance and hence the recall is 100%, making precision and accuracy equivalent. The accuracy of the majority classifier, which always guesses the intended sense to be the majority sense in the training data, is provided as a point of comparison. In the case of Senseval-1 and Senseval-2 data, a break down of the accuracies for each part of speech is depicted as well. It may be noted that we performed experiments with parts of speech of words in close vicinity (at most 2 words away from the target word) to the target word as part of speech features have a highly localized effect.

We observe that that except for *line* and *hard* data, the individual word part of speech have performed better than the majority sense classifier. In the case of *serve* and *interest* data their performance is significantly better than the majority sense baseline for almost all the part of speech features considered. A striking pattern that emerges from these results is the high accuracies achieved using the part of speech of the word immediately following the target word (P_1). Except for *line* and *hard* data it is found to give the best results. The break down of accuracies for individual parts of speech helps explain this phenomenon. In the case of

Table 40: Accuracy using Individual Part of Speech Features on Senseval-2 Data.

	All	**Nouns**	**Verbs**	**Adjectives**
Majority Classifier	47.7%	51.0%	39.7%	59.0%
P_{-2}	47.1%	51.9%	38.0%	57.9%
P_{-1}	49.6%	**55.2%**	40.2%	59.0%
P_0	49.9%	**55.7%**	40.6%	58.2%
P_1	**53.1%**	**53.8%**	**49.1%**	**61.0%**
P_2	48.9%	50.2%	43.2%	59.4%

Table 41: Accuracy using Individual Part of Speech Features on Senseval-1 Data.

	All	**Nouns**	**Verbs**	**Adjectives**	**Indeterminates**
Majority Classifier	56.3%	57.2%	56.9%	64.3%	43.8%
P_{-2}	57.5%	58.2%	58.6%	64.0%	48.9%
P_{-1}	59.2%	**62.2%**	58.2%	64.3%	51.8%
P_0	60.3%	**62.5%**	58.2%	64.3%	57.1%
P_1	**63.9%**	**58.2%**	**58.6%**	**64.0%**	48.9%
P_2	59.9%	60.0%	**60.8%**	**65.2%**	53.4%

Table 42: Accuracy using Individual Part of Speech Features on *line, hard, serve* and *interest* Data.

	line	**hard**	**serve**	**interest**
Majority Classifier	54.3%	81.5%	42.2%	54.9%
P_{-2}	54.9%	81.6%	52.1%	56.0%
P_{-1}	**56.2%**	82.1%	54.8%	62.7%
P_0	**54.3%**	81.6%	47.4%	64.0%
P_1	**54.2%**	81.6%	**55.6%**	**65.3%**
P_2	54.3%	81.7%	48.9%	62.3%

verbs and adjectives, the disambiguation accuracy using part of speech tags of words at one or two positions to the right of the target word is the highest. We believe that this occurs as words in these positions may act as objects to target words which are verbs (For example, *drink water*). Similarly, when the target word is an adjective, the words immediately to its right are possibly nouns which are qualified by the adjective (For example, *short discussion*. As words immediately to the right of the target word may have strong syntactic relations with the target word, their parts of speech are strongly suggestive of the intended sense. By the same reasoning, nouns will be expected to be disambiguated best by part of speech of words on its immediate left. However, we find that nouns are helped by the target word part of speech (P_0) and the parts of speech of words adjacent to it on either side (P_{-1}, P_1). This is justified as nouns act as subjects in a sentence and a subject is followed by a syntactically related verb. Thus words on either side of nouns bear important syntactic relations and are hence equivalently suggestive at its intended sense. A sample decision tree learnt is shown in figure 46

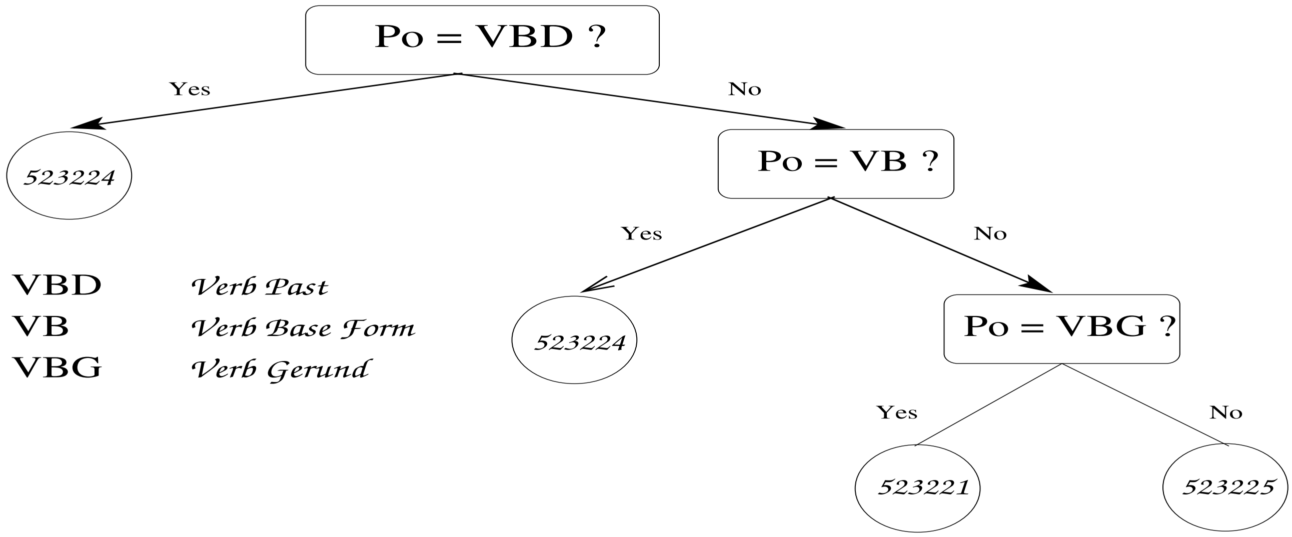

Figure 46: Sample Decision Tree learnt using Individual Word POS as features

4.1.3 Parse Features

Experiments were performed to evaluate the accuracy achieved with various parse features. Tables 43, 44, and 45 show the results using individual parse features on Senseval-2, Senseval-1 and *line, hard, serve* and *interest* data, respectively. The recall is 100% throughout. The accuracy of the majority classifier is specified as a point of comparison. In case of Senseval-1 and Senseval-2 data, a break down of the accuracies for each part of speech is shown as well.

Table 43: Accuracy using Parse Features on Senseval-2 Data.

	All	**Nouns**	**Verbs**	**Adjectives**
Majority Classifier	47.7%	51.0%	39.7%	59.0%
Head Word	**51.7%**	**58.5%**	39.8%	**64.0**
Head of Parent Phrase	50.0%	56.1%	40.1%	59.3
Phrase (POS)	48.3%	51.7%	40.3%	59.5
Parent Phrase (POS)	48.5%	53.0%	39.1%	60.3

Table 44: Accuracy using Parse Features on Senseval-1 Data.

	All	**Nouns**	**Verbs**	**Adjectives**	**Indeterminates**
Majority Classifier	56.3%	57.2%	56.9%	64.3%	43.8%
Head Word	**64.3%**	**70.9%**	59.8%	**66.9%**	59.7%
Head of Parent Phrase	60.6%	62.6%	60.3%	65.8%	53.4%
Phrase (POS)	58.5%	57.5%	57.2%	66.2%	55.2%
Parent Phrase (POS)	57.9%	58.1%	58.3%	66.2%	50.0%

The head word of the phrase housing the target word, or head word for short, has produced the most accurate results in almost all the data. The results for nouns and adjectives are most improved by this features. We believe this is linguistically justified as the head word of a phrase is a content word and nouns in a sentence are generally associated with other content words in the context. In case of adjectives the relation is likely to be even stronger as the head word is likely to be the noun being qualified by the adjective. The head of parent phrase is found to be very useful in the *line data*. Content words like *draw, cross* and *write* which were strongly associated with the sense of the target word *line* were picked up as the head of parent phrase. The phrase of the target word and its parent phrase are not found to be useful when used as the lone features. We believe this is due to the fact that a target word occurs in a handful of phrases and thus the tree learnt

Table 45: Accuracy using Parse Features on *line, hard, serve* and *interest* Data.

	line	**hard**	**serve**	**interest**
Majority Classifier	54.3%	81.5%	42.2%	54.9%
Head Word	54.7%	**87.8%**	47.4%	**69.1%**
Head of Parent Phrase	**59.8%**	84.5%	57.2%	67.8%
Phrase (POS)	54.3%	81.5%	41.4%	54.9%
Parent Phrase (POS)	54.3%	81.7%	41.6%	54.9%

is very close to a majority classifier. Figures 47 and 48 depict sample decision trees learnt using head word and phrase of the target word as features.

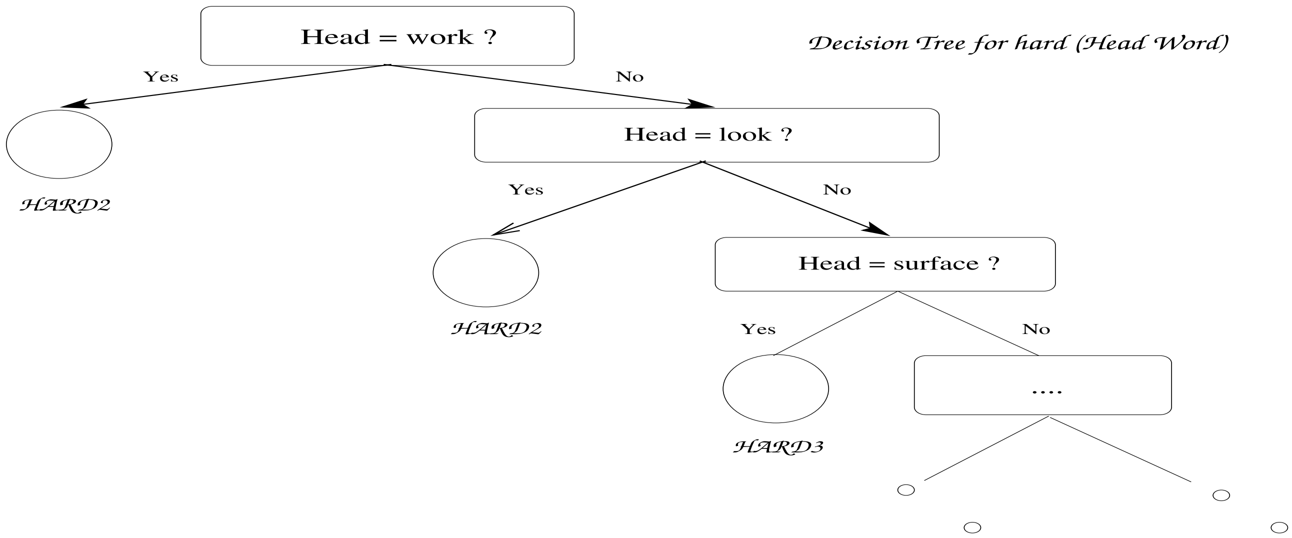

Figure 47: Sample Decision Tree learnt using Head Word as features

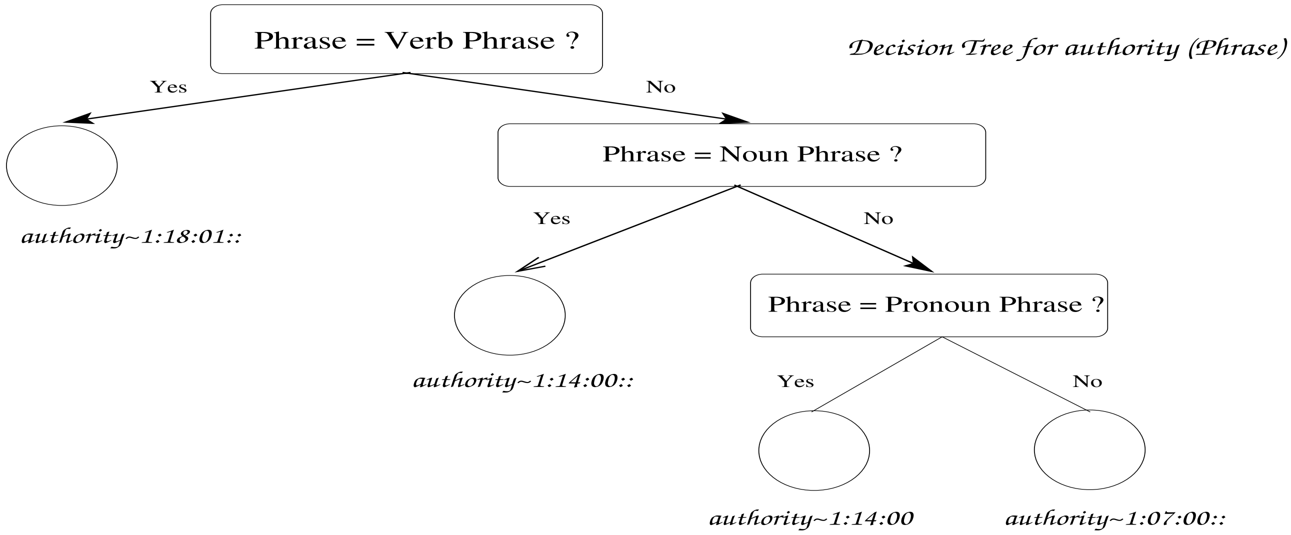

Figure 48: Sample Decision Tree learnt using Phrase as features

4.2 Complementarity and Redundancy

We have seen a rich set of features which may be utilized for word sense disambiguation. It is expected that a certain number of instances which are correctly tagged by one feature X, are correctly disambiguated by another feature Y, as well. That is to say, there is a certain amount of redundancy amongst X and Y. On the other hand, it is expected that some number of instances are correctly disambiguated by X and not Y, and vice versa. That is, the features X and Y are complementary to a certain extent. In order to gain an insight into the amount of complementarity and redundancy amongst two features or two sets of features, we introduce a few measures to compare the sense disambiguations done by two different features. The *Dice Agreement* is used to quantify the similarity in tagging as done by the two features. For each instance, the similarity in tagging by the two features is calculated by using the Dice Coefficient. Let A be the set containing the senses assigned to the instance by feature X. Similarly, let B be the set containing the senses assigned to the instance by feature Y. Then the Dice Coefficient may be calculated as follows:

$$Dice = \frac{2 \, * \, |A \, \cap \, B|}{|A| \, + \, |B|} \tag{22}$$

If feature set 1 tags it with senses l and m while feature 2 tags it l, n and p, the Dice Coefficient is 2 x 1 / (2 + 3) = .4. This similarity is calculated for each instance and the average value will be referred to as

Dice Agreement. Given two features, we calculate the *Optimal Ensemble* and *Baseline Ensemble*, of the two features. By *Optimal Ensemble*, we mean the accuracy attained by a hypothetical ensemble which predicts the intended sense correctly if either of the two individual features suggests the correct sense. If neither of the feature sets suggests the desired sense, then the ensemble fails to get the intended sense. Such an ensemble, albeit hypothetical, provides an upper bound of the accuracy achievable by combining the individual features. *Baseline Ensemble* is again a hypothetical ensemble useful in quantifying the redundancy in the two features or feature sets. By redundancy we mean the amount by which the discrimination knowledge provided by feature set 1 is provided by feature set 2 as well. The ensemble correctly predicts the intended sense only when both individual features suggest the correct sense. In case any of the feature sets suggests multiple senses, the ensemble identifies the senses in common as the intended sense. Two additional measures are provided which shed light into the redundant instances (instances correctly tagged by both feature sets). *Absolute Agreement*, or *Agreement* for short, is the ratio of the redundant instances to the total instances. Let N be the number of instances being disambiguated. Let M be the instances which are assigned at least one sense in common by both feature X and Y. Then the *Absolute Agreement* between X and Y is defined to be:

$$Absolute\ Agreement = \frac{M}{N} \tag{23}$$

Precision on Agreement is the ratio of the redundant instances which are correctly tagged to the redundant instances. As defined earlier, let A be the set of senses assigned to an instance by feature X. Similarly, let B be the set of senses assigned to the instance by feature Y. Let C be the number of senses from the intersection of A and B, which are correct. Then the precision on agreement for the instance is calculated as follows:

$$Precision\ on\ Agreement\ per\ Instance\ = \frac{|A| \cap |B|}{C} \tag{24}$$

The over all *Precision on Agreement* is calculated by averaging the Precision on Agreement for each of the M instances which are assigned at least one sense in common by both X and Y. It may be noted that *Baseline Ensemble* is equal to the product of *Absolute Agreement* and *Precision on Agreement*.

$$Baseline\ Ensemble\ =\ Absolute\ Agreement\ *\ Precision\ on\ Agreement \tag{25}$$

We also note that the difference of accuracies attained by individual features (Accuracy X, Accuracy Y, say) with the *Baseline Ensemble* added to the *Baseline Ensemble* equals the *Optimal Ensemble*.

$$Optimal\ Ensemble\ =\ Baseline\ Ensemble\ +\ (Accuracy\ X\ -\ Baseline\ Ensemble)$$

$$+ \, (Accuracy \, Y \; - \; Baseline \; Ensemble \,) \qquad (26)$$

The higher the difference between the *Optimal Ensemble* and the accuracy achieved by the better of the two features, the more there is reason to combine the two features. Also, higher the Baseline Ensemble more is the redundancy amongst the features.

4.2.1 Complementarity and Redundancy amongst Lexical Features

Tables 46 and 47 show the complementarity and redundancy in discriminating information provided by the various lexical features. The accuracies attained by individual features are given next to them to serve as a point of comparison. We observe that Unigrams and Bigrams, albeit similar in nature, are significantly complementary and the overall accuracy may be improved by their combination.

Table 46: Redundancy and Complementarity amongst Lexical Features in Senseval-2 Data.

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
Surface Form	49.3%	Unigrams	55.3%	73.0%	61.1%	45.6% (77.0 * 59.2)
Unigrams	55.3%	Bigrams	55.1%	73.8%	63.9%	48.2% (76.9 * 62.7)
Bigrams	55.1%	Surface Form	49.3%	74.6%	60.0%	45.7% (77.6 * 58.9)

Table 47: Redundancy and Complementarity amongst Lexical Features in Senseval-1 Data.

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
Surface Form	62.9%	Unigrams	66.9%	81.5%	71.8%	59.9% (84.9 * 70.6)
Unigrams	66.9%	Bigrams	66.9%	76.7%	74.9%	59.6% (78.3 * 76.1)
Bigrams	66.9%	Surface Form	62.9%	76.1%	73.0%	58.3% (78.4 * 74.3)

4.2.2 Complementarity and Redundancy amongst Syntactic Features

In order to gain an insight into the complementarity and redundancy of the individual part of speech features we compared the tagging as done by different feature sets. Table 48 and 49 depict our findings for Senseval-2 and Senseval-1 data. The accuracies attained by individual features are given next to them to serve as a point of comparison.

Table 48: Part of Speech Feature Redundancy and Complementarity in Senseval-2 Data

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
P_{-1}	49.6%	P_0	49.9%	72.9%	57.5%	44.4% (78.4 * 56.6)
P_0	49.9%	P_1	53.1%	67.1%	61.1%	43.3% (71.1 * 60.9)
P_1	53.1%	P_2	48.9%	69.5%	59.8%	44.2% (74.0 * 59.7)
P_0, P_1	54.3%	P_{-1}	49.6%	62.1%	62.9%	43.5% (67.2 * 64.8)
P_{-1}, P_0, P_1	54.6%	P_{-2}	47.1%	58.1%	63.0%	40.3% (61.9 * 65.0)
P_{-1}, P_0, P_1	54.6%	P_2	48.9%	60.4%	63.3%	41.7% (64.0 * 65.1)

Table 49: Part of Speech Feature Redundancy and Complementarity in Senseval-1 Data

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
P_{-1}	59.2%	P_0	60.3%	77.6%	66.9%	54.4% (81.5 * 66.7)
P_0	60.3%	P_1	63.9%	74.9%	69.8%	56.3% (78.2 * 71.9)
P_1	63.9%	P_2	59.9%	78.2%	68.8%	56.5% (81.2 * 69.5)
P_0, P_1	66.7%	P_{-1}	59.2%	70.6%	72.4%	54.8% (73.4 * 74.7)
P_{-1}, P_0, P_1	68.0%	P_{-2}	57.5%	67.6%	73.6%	53.7% (71.1 * 75.5)
P_{-1}, P_0, P_1	68.0%	P_2	59.9%	71.8%	73.2%	56.3% (74.9 * 74.2)

The parts of speech of individual words show a certain amount of complementarity amongst each other. The large differences between optimal accuracies of such combinations compared to the individual accuracies suggests that there will be better disambiguation using such pairs of features for disambiguation rather than just one. The part of speech combination of the target word and the word to its right (P_0, P_1) in general has high *Optimal Ensemble* and so is most suitable for combination. The similarity in the assignments based on P_0-P_1 pair is significantly different from that of P_{-1} as is seen by their relatively lower similarity values and once again there seems to be benefit in combining the three features based on the higher optimal accuracy values for the (P_0, P_1)-(P_{-1}) pair. We also observe that the feature set pairs (P_{-1}, P_0, P_1) - (P_{-2}) and (P_{-1}, P_0, P_1) - (P_2) do not have much higher optimal accuracies than that of (P_{-1}, P_0, P_1) itself. This suggests that using the part of speech tags of words at two positions to the left and right of the target word, in addition to (P_{-1}, P_0, P_1) is not likely to gain much.

Based on observed results for part of speech features and the complementary redundancy values two sets of part of speech features stand out as good candidates to be used for word sense disambiguation - (P_0, P_1) and (P_{-1}, P_0, P_1)

4.2.3 Complementarity and Redundancy across Lexical and Syntactic Features

Tables 50 through tab:Redundancy Across interest depict the redundancy and complementarity across the best of the lexical and syntactic features identified in the previous sections. Note, we compare additional pairs for Senseval-2 data.

The additional comparisons in the Senseval-2 data (table 50 are meant to study some of the part of speech features in combination with lexical features. We observe that the various individual part of speech features show varying similarities with lexical features (Unigrams and Bigrams). The P_1 feature stands out as being least similar (low Dice Agreement) with the lexical features and hence having greater potential of being complementary to them. The high values of *Optimal Ensemble* with P_1 identifies it as being more complementary with the lexical features than other individual parts of speech.

In general we observe a good deal of complementarity across lexical and syntactic features in all the data. This may be deduced from the high values of *Optimal Ensemble* as compared to individual accuracies. The complementarity is markedly less in *hard* data for which the individual features itself attain a very high accuracy. We also note that the *Optimal Ensemble* in general is lower across part of speech features parse features than in case of part of speech features and lexical features. this suggests that the latter pair is more

Table 50: Redundancy and Complementarity Across Knowledge Sources in Senseval-2 Data.

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
Unigrams	55.3%	P_0	49.9%	69.9%	61.9%	45.1% (73.7 * 61.2)
Unigrams	55.3%	P_{-1}	49.6%	64.7%	63.6%	43.9% (69.5 * 63.1)
Unigrams	55.3%	$\mathbf{P_1}$	53.1%	**63.2%**	**66.2%**	44.2% (67.1 * 65.8)
Unigrams	55.3%	P_2	48.9%	64.0%	63.3%	42.8% (67.9 * 63.1)
Unigrams	55.3%	$\mathbf{P_0, P_1}$	54.3%	62.0%	67.1%	44.4% (65.9 * 67.3)
Unigrams	55.3%	P_{-1}, P_0, P_1	54.6%	59.4%	67.9%	43.6% (62.7 * 69.4)
Bigrams	55.1%	P_0	49.9%	71.9%	61.0%	45.2% (75.1 * 60.2)
Bigrams	55.1%	P_{-1}	49.6%	70.1%	61.4%	45.5% (74.6 * 61.0)
Bigrams	55.1%	P_1	53.1%	69.4%	63.4%	46.3% (72.8 * 63.6)
Bigrams	55.1%	P_2	48.9%	72.3%	61.2%	44.5% (72.3 * 61.5)
Bigrams	55.1%	$\mathbf{P_{-1}, P_0, P_1}$	54.6%	**62.4%**	**66.0%**	44.7% (65.3 * 68.5)
Unigrams	55.3%	Head	51.7%	70.8%	62.9%	46.4% (75.2 * 61.7)
Unigrams	55.3%	Parent	50.0%	66.3%	63.1%	43.4% (69.4 * 62.5)
Unigrams	55.3%	Head, Parent	52.6%	67.4%	64.3%	45.4% (71.1 * 63.8)
Bigrams	55.1%	Head	51.7%	73.5%	61.5%	46.9% (76.9 * 61.0)
Bigrams	55.1%	Parent	50.0%	72.8%	60.3%	45.5% (75.4 * 60.4)
Bigrams	55.1%	Head, Parent	52.6%	71.8%	62.2%	46.8% (74.8 * 62.5)
P_0, P_1	54.3%	Head	51.7%	69.3%	62.0%	45.9% (74.4 * 61.7)
P_0, P_1	54.3%	Parent	50.0%	64.7%	62.4%	43.1% (68.2 * 63.2)
P_0, P_1	54.3%	Head, Parent	52.6%	67.8%	**62.8%**	45.6% (72.0 * 63.4)
P_{-1}, P_0, P_1	54.6%	Head, Parent	54.6%	64.6%	**63.9%**	44.7% (68.4 * 65.4)

Table 51: Redundancy and Complementarity Across Knowledge Sources in Senseval-1 Data.

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
Unigrams	66.9%	P_0, P_1	66.7%	71.8%	76.6%	58.1% (74.2 * 78.3)
Unigrams	66.9%	P_{-1}, P_0, P_1	68.0%	69.4%	78.0%	57.6% (71.7 * 80.4)
Bigrams	66.9%	P_0, P_1	66.7%	75.1%	74.9%	59.2% (77.0 * 76.9)
Bigrams	66.9%	P_{-1}, P_0, P_1	68.0%	75.2%	75.9%	59.5% (75.2 * 79.1)
Unigrams	66.9%	Head, Parent	65.1%	75.6%	74.8%	58.2% (77.9 * 74.6)
Bigrams	66.9%	Head, Parent	65.1%	76.1%	74.6%	58.3% (78.0 * 74.8)
P_0, P_1	66.7%	Head, Parent	65.1%	76.3%	73.7%	59.0% (78.9 * 74.9)
P_{-1}, P_0, P_1	68.0%	Head, Parent	65.1%	73.0%	75.4%	58.8% (75.5 * 77.9)

Table 52: Redundancy and Complementarity Across Knowledge Sources in *line* Data.

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
Unigrams	74.5%	P_0, P_1	54.1%	69.4	77.2%	53.1% (71.9 * 73.9)
Unigrams	74.5%	P_{-1}, P_0, P_1	60.4%	65.2%	82.0%	55.1% (68.3 * 80.7)
Bigrams	72.9%	P_0, P_1	54.1%	75.6%	74.4%	53.9% (77.4 * **60.7**)
Bigrams	72.9%	P_{-1}, P_0, P_1	60.4%	68.5%	79.2%	55.9% (70.7 * 79.1)
Unigrams	74.5%	Head, Parent	60.4%	69.5%	80.1%	55.6% (70.8 * 78.5)
Bigrams	72.9%	Head, Parent	60.4%	73.2%	78.0%	55.5% (73.7 * 75.3)
P_0, P_1	54.1%	Head, Parent	60.4%	86.2%	62.5%	52.6% (87.7 * 60.0)
P_{-1}, P_0, P_1	60.4%	Head, Parent	60.4%	74.2%	70.4%	51.6% (76.5 * 67.4)

Table 53: Redundancy and Complementarity Across Knowledge Sources in *hard* Data.

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
Unigrams	83.4%	P_0, P_1	81.9%	92.7%	86.4%	79.6% (93.4 * 85.2)
Unigrams	83.4%	P_{-1}, P_0, P_1	84.8%	87.7%	89.7%	79.4% (88.6 * 89.6)
Bigrams	89.5%	P_0, P_1	81.9%	90.2%	90.4%	81.1% (90.3 * 89.8)
Bigrams	89.5%	P_{-1}, P_0, P_1	84.8%	89.3%	91.8%	82.7% (89.5 * 92.4)
Unigrams	83.4%	Head, Parent	87.7%	89.5%	90.7%	81.1% (90.3 * 89.9)
Bigrams	89.5%	Head, Parent	87.7%	94.2%	91.3%	86.1% (94.5 * 91.1)
P_0, P_1	81.9%	Head, Parent	87.7%	91.5%	88.8%	80.8% (91.6 * 88.2)
P_{-1}, P_0, P_1	84.8%	Head, Parent	87.7%	88.1%	91.4%	81.2% (88.4 * 91.9)

Table 54: Redundancy and Complementarity Across Knowledge Sources in *serve* Data.

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
Unigrams	73.2%	P_0, P_1	60.2%	60.1%	85.1%	58.4% (69.6 * 83.9)
Unigrams	73.2%	P_{-1}, P_0, P_1	73.0%	60.8%	89.9%	58.4% (62.5 * 93.5)
Bigrams	72.1%	P_0, P_1	60.2%	69.1%	81.2%	61.7% (61.7 * 78.6)
Bigrams	72.1%	P_{-1}, P_0, P_1	73.0%	69.7%	85.6%	60.9% (71.4 * 85.3)
Unigrams	73.2%	Head, Parent	58.1%	53.7%	84.4%	47.6% (54.5 * 87.2)
Bigrams	72.1%	Head, Parent	58.1%	60.2%	80.2%	50.7% (61.1 * 83.0)
P_0, P_1	60.2%	Head, Parent	58.1%	56.3%	77.8%	50.6% (65.2 * 77.5)
P_{-1}, P_0, P_1	73.0%	Head, Parent	58.1%	59.2%	81.8%	49.9% (59.9 * 83.4)

Table 55: Redundancy and Complementarity Across Knowledge Sources in *interest* Data.

Feature-Set Pair				**Dice**	**Optimal**	**Baseline Ensemble**
Set 1	Accuracy	Set2	Accuracy	**Agree.**	**Ens.**	**(Agree. * Prec. on Agree.)**
Unigrams	75.7%	P_0, P_1	70.5%	63.2%	88.5%	59.4% (64.6 * 91.9)
Unigrams	75.7%	P_{-1}, P_0, P_1	78.8%	66.0%	91.47%	65.3% (67.6 * **96.6**)
Bigrams	79.9%	P_0, P_1	70.5%	68.0%	87.9%	63.2% (68.6 * 92.0)
Bigrams	79.9%	P_{-1}, P_0, P_1	78.8%	70.3%	90.1%	69.5% (71.4 * 97.4)
Unigrams	75.7%	Head, Parent	61.8%	61.8%	90.5%	59.2% (62.3 * 94.9)
Bigrams	79.9%	Head, Parent	61.8%	65.7%	90.4%	62.3% (65.9 * 95.5)
P_0, P_1	70.5%	Head, Parent	61.8%	83.8%	77.7%	66.3% (84.6 * 78.4)
P_{-1}, P_0, P_1	78.8%	Head, Parent	61.8%	78.7%	85.6%	67.6% (79.8 * 84.7)

complementary than the former.

We note that the *Agreement* values amongst the feature sets lies between .55 and .75 except for *hard* data for which we would expect much higher agreements due to the high accuracies. We would expect these values to be above the accuracy of the least accurate feature set in the ensemble. This is because, if both systems with accuracies greater than .5 suggest the same sense then it is more likely that the suggested sense is correct than the accuracy of the worst system in the ensemble. Except for the bigram-POS(P_0, P_1) combination in *line* data, we observe this inequality to be true. We would always like the precision on agreement values to be high, since both systems are suggesting the same sense in these instances. However, we would like the feature sets being combined to have low similarity, this is so that they may be good at correctly tagging different kinds of instances and hence a good ensembling technique may be useful in attaining high accuracies. If the agreement is very high, even the best ensembling technique will not help much more than the individual components. Thus, two summarize two individual feature sets are worth combining with an ensemble technique if, their agreement is low while the precision on agreement and *Optimal Ensemble* are high. In the comparisons done, we note that of all the individual part of speech features, P_1 is most suitable for combining with lexical features. We observe that there is significant complementarity across lexical and syntactic features. We also conclude that part of speech and lexical features are more complementary than part of speech and parse features.

4.3 Combining Features

4.3.1 Sequences of Parts of Speech

Experiments were conducted using part of speech sequences corresponding to a sequence of words, as features. This gives additional importance to a sequence of part of speech tags. We believe that better results may be obtained this way if part of speech sequences are suggestive of the intended sense. Results on using sequences of parts of speech are shown in Tables 56, 57 and 58.

Table 56: Accuracy using Part of Speech Sequences on Senseval-2 Data.

	All	**Nouns**	**Verbs**	**Adjectives**
Majority Classifier	47.7%	51.0%	39.7%	59.0%
$\mathbf{P_{-2}P_{-1}}$	48.9%	53.8%	39.7%	59.0%
$\mathbf{P_{-1}P_0}$	50.8%	**57.8%**	40.4%	59.1%
$\mathbf{P_0P_1}$	**53.8%**	**57.2%**	**47.7%**	**60.5%**
$\mathbf{P_1P_2}$	**52.4%**	53.6%	**48.1%**	59.5%
$\mathbf{P_{-1}P_1}$	**51.9%**	55.3%	45.3%	59.5%
$\mathbf{P_0P_1P_2}$	51.4%	54.6%	44.9%	59.5%
$\mathbf{P_{-1}P_0P_1}$	**52.0%**	56.8%	44.1%	59.3%

Part of speech sequences have in general performed better than individual part of speech features, indicating an amount of complementarity in the discrimination information provided by the individual tags. The sequences involving the word to the right of the target word (P_1) show best results. The sequence $P_{-1}P_1$ is found to be almost as good as $P_{-1}P_0P_1$ in most data. It may be noted that all target words for a given task belong to the same broad part of speech such as noun, verb and adjective. Thus, P_0 is useful only if the subtle distinctions within nouns, verbs and adjectives are suggestive of the intended sense. Figure 49 depicts a sample decision tree learnt using sequence of part of speech.

Table 57: Accuracy using Part of Speech Sequences on Senseval-1 Data.

	All	**Nouns**	**Verbs**	**Adjectives**	**Indeterminates**
Majority Classifier	56.3%	57.2%	56.9%	64.3%	43.8%
$\mathbf{P}_{-2}\mathbf{P}_{-1}$	58.9%	62.1%	58.3%	64.4%	50.5%
$\mathbf{P}_{-1}\mathbf{P}_{0}$	61.6%	**65.8%**	59.7%	64.4%	56.3%
$\mathbf{P}_{0}\mathbf{P}_{1}$	**65.5%**	**66.5%**	**66.3%**	**66.2%**	**62.0%**
$\mathbf{P}_{1}\mathbf{P}_{2}$	62.5%	**63.7%**	65.7%	57.3%	57.3%
$\mathbf{P}_{-1}\mathbf{P}_{1}$	**62.2%**	64.4%	63.4%	65.0%	54.7%
$\mathbf{P}_{0}\mathbf{P}_{1}\mathbf{P}_{2}$	62.8%	64.3%	63.0%	65.7%	57.7%
$\mathbf{P}_{-1}\mathbf{P}_{0}\mathbf{P}_{1}$	**63.3%**	66.8%	63.3%	65.0%	56.9%

Table 58: Accuracy using Part of Speech Sequences on *line, hard, serve* and *interest* Data.

	line	**hard**	**serve**	**interest**
Majority Classifier	54.3%	81.5%	42.2%	54.9%
$\mathbf{P}_{-2}\mathbf{P}_{-1}$	**57.3%**	83.2%	59.4%	62.6%
$\mathbf{P}_{-1}\mathbf{P}_{0}$	**56.5%**	82.1%	59.9%	66.9%
$\mathbf{P}_{0}\mathbf{P}_{1}$	54.2%	81.8%	59.2%	69.5%
$\mathbf{P}_{1}\mathbf{P}_{2}$	**56.1%**	82.9%	58.1%	67.2%
$\mathbf{P}_{-1}\mathbf{P}_{1}$	58.6%	83.9%	**67.2%**	**72.0%**
$\mathbf{P}_{0}\mathbf{P}_{1}\mathbf{P}_{2}$	55.8%	82.8%	61.5%	68.5%
$\mathbf{P}_{-1}\mathbf{P}_{0}\mathbf{P}_{1}$	60.1%	83.9%	69.7%	73.8%

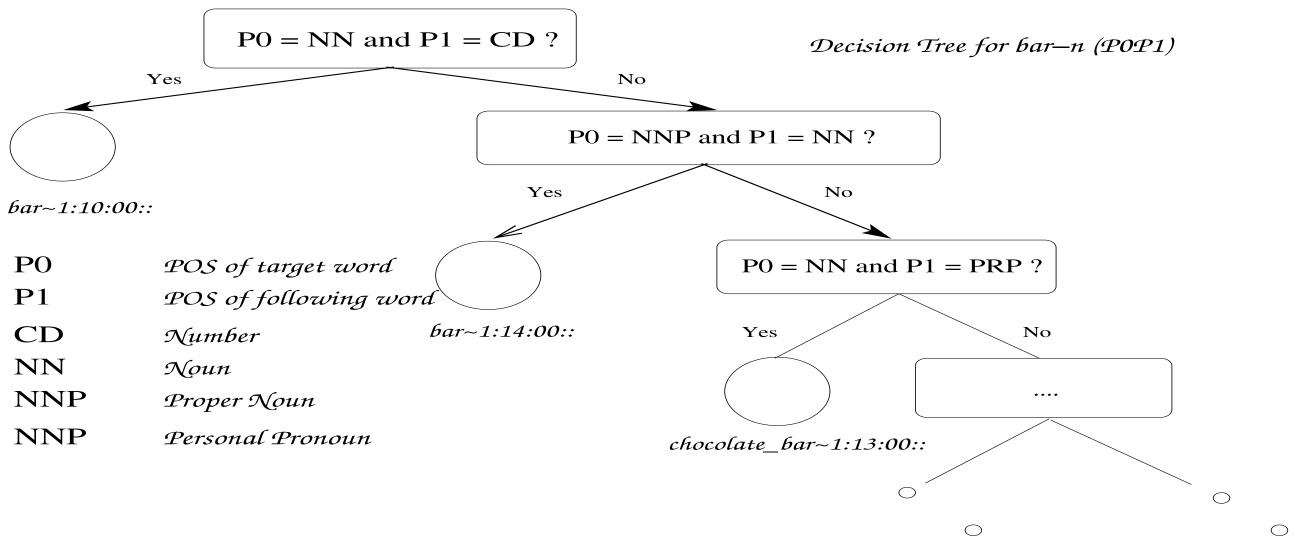

Figure 49: Sample Decision Tree learnt using Sequence of POS as features

4.3.2 Combination of Part of Speech features

The context around the target word is captured to a greater extent by a combination of the many part of speech features discussed so far. Tables 59, 60 and 61 depict the results attained using combinations of individual word and sequences of part of speech tags. Note, here a single decision tree is created with multiple part of speech features.

Combinations of part of speech features have outperformed both individual word part of speech features and sequential part of speech features. This once again suggests a complementarity amongst individual word part of speech features and also that combination of individual part of speech features better captures this complementarity than sequences of tags. The combinations including P_1 tag continue to get best results. In Senseval-1 and Senseval-2 data, we observe that verbs and adjectives give best results with the P_0, P_1 combination which as we pointed out earlier, is likely to capture the verb-object and adjective-noun relations. We observe that nouns are disambiguated best using features from either side (P_{-1}, P_0, P_1). *line, hard, serve* and *interest* data deviate from the behavior of Senseval-1 and Senseval-2 data in that the two nouns, verb and adjective are disambiguated best using a broad window of parts of speech of two words on either side of the target word along with that of the target word itself (P_{-2}, P_{-1}, P_0, P_1, P_2). We believe that this is primarily due to the large amount of test and training data associated with these words. Words further away

Table 59: Accuracy in Part of Speech Combinations on Senseval-2 Data.

	All	**Nouns**	**Verbs**	**Adjectives**
Majority Classifier	47.7%	51.0%	39.7%	59.0%
P_{-1}, P_0	50.8%	58.3%	40.1%	58.7%
P_0, P_1	**54.3%**	57.5%	**48.3%**	**60.8%**
P_1, P_2	53.2%	53.8%	49.7%	60.0%
P_{-1}, P_0, P_1	**54.6%**	59.8%	47.4%	59.8%
P_{-2}, P_{-1}, P_0, P_1, P_2	54.6%	**60.3%**	47.6%	58.3%
$P_{-1}P_0$, P_0P_1	54.0%	59.3%	46.2%	60.0%
P_0, $P_{-2}P_{-1}$, P_1P_2	53.7%	58.4%	46.7%	59.2%

Table 60: Accuracy using Part of Speech Combinations on Senseval-1 Data.

	All	**Nouns**	**Verbs**	**Adjectives**	**Indeterminates**
Majority Classifier	56.3%	57.2%	56.9%	64.3%	43.8%
P_{-1}, P_0	62.2%	67.0%	58.0%	64.5%	60.1%
P_0, P_1	**66.7%**	68.7%	**66.5%**	**66.2%**	64.7%
P_1, P_2	64.0%	65.6%	64.6%	65.8%	59.2%
P_{-1}, P_0, P_1	**68.0%**	72.8%	66.1%	65.6%	66.5%
P_{-2}, P_{-1}, P_0, P_1, P_2	67.8%	**73.1%**	66.0%	63.9%	66.1%
$P_{-1}P_0$, P_0P_1	66.7%	70.8%	65.6%	65.6%	63.6%
P_0, $P_{-2}P_{-1}$, P_1P_2	66.6%	68.3%	65.6%	66.7%	65.8%

Table 61: Accuracy using Part of Speech Combinations on *line, hard, serve* and *interest* Data.

	line	**hard**	**serve**	**interest**
Majority Classifier	54.3%	81.5%	42.2%	54.9%
P_{-1}, P_0	56.5%	82.3%	60.3%	67.7%
P_0, P_1	54.1%	81.9%	60.2%	70.5%
P_1, P_2	55.9%	82.2%	58.0%	68.6%
P_{-1}, P_0, P_1	60.4%	84.8%	73.0%	78.8%
P_{-2}, P_{-1}, P_0, P_1, P_2	**62.3%**	**86.2%**	**75.7%**	**80.6%**
$P_{-1}P_0$, P_0P_1	60.2%	84.8%	70.7%	79.1%
P_0, $P_{-2}P_{-1}$, P_1P_2	**63.1%**	85.8%	73.0%	77.8%

from the target word tend to be weak indicators of the intended sense and may easily overcome by spurious instances. In case of Senseval-1 and Senseval-2 data, the lack of sufficient data might have prevented such features from being helpful.

4.3.3 Guaranteed Pre-Tagging

We use Guaranteed pre-Tagging [42] in the part of speech tagging of the text. Experiments were conducted to compare the results of word sense disambiguation and see if guaranteed pre-tagging has helped better disambiguation. Tables 62 and 63 show the results with and without guaranteed pre-tagging. It may be noted that Mohammad and Pedersen [42] point out that the Brill Tagger derived its rules automatically from the wall street journal corpus and so the number of contextual rules triggered by Senseval-2 data are very low. Hence the effect of guaranteed pre-tagging to show up in the learned decision trees is expected to be sparse.

We observe that for most of the features there is an improvement with guaranteed pre-tagging. The ones for which there is a decline in accuracy are italicized. We believe that with a better rule set for the tagger when the contextual tagger plays a more dominant role, the significance of pre-tagging will be even more.

Table 62: Effect of Guaranteed Pre-Tagging on WSD as on Senseval-2 data

	Guaranteed Pre-Tagging	**Regular Pre-Tagging**
Majority Classifier	47.7%	47.7%
P_{-1}	49.6%	49.8%
P_0	49.9%	49.5%
P_1	53.1%	53.1%
$\mathrm{P}_{-2}\mathrm{P}_{-1}$	48.9%	48.4%
$P_{-1}P_0$	50.8%	51.2%
$\mathrm{P}_0\mathrm{P}_1$	53.8%	53.1%
$\mathrm{P}_1\mathrm{P}_2$	52.4%	51.1%
$\mathrm{P}_{-1}\mathrm{P}_1$	51.9%	51.1%
$\mathrm{P}_0\mathrm{P}_1\mathrm{P}_2$	51.4%	50.6%
$\mathrm{P}_{-1}\mathrm{P}_0\mathrm{P}_1$	52.0%	51.7%
P_{-1}, P_0	50.8%	50.9%
P_0, P_1	54.3%	53.8%
P_{-1}, P_0, P_1	54.6%	54.7%
P_{-2}, P_{-1}, P_0, P_1, P_2	54.6%	54.1%
$\mathrm{P}_{-1}\mathrm{P}_0$, $\mathrm{P}_0\mathrm{P}_1$	54.0%	53.7%
P_0, $\mathrm{P}_{-2}\mathrm{P}_{-1}$, $\mathrm{P}_1\mathrm{P}_2$	53.7%	52.4%

Table 63: Effect of Guaranteed Pre-Tagging on WSD as on Senseval-1 data

	Guaranteed Pre-tagging	**Regular Pre-tagging**
Majority Classifier	56.3%	56.3%
P_{-1}	59.2%	59.5%
P_0	60.3%	60.0%
P_1	63.9%	63.6%
$\mathrm{P}_{-2}\mathrm{P}_{-1}$	58.9%	59.1%
$\mathrm{P}_{-1}\mathrm{P}_0$	61.6%	61.4%
$\mathrm{P}_0\mathrm{P}_1$	65.5%	65.0%
$\mathrm{P}_1\mathrm{P}_2$	62.5%	62.4%
$\mathrm{P}_{-1}\mathrm{P}_1$	62.2%	62.2%
$\mathrm{P}_0\mathrm{P}_1\mathrm{P}_2$	62.8%	62.7%
$\mathrm{P}_{-1}\mathrm{P}_0\mathrm{P}_1$	63.3%	63.2%
P_{-1}, P_0	62.2%	62.1%
P_0, P_1	66.7%	66.7%
P_{-1}, P_0, P_1	68.0%	67.6%
P_{-2}, P_{-1}, P_0, P_1, P_2	67.8%	66.1%
$\mathrm{P}_{-1}\mathrm{P}_0$, $\mathrm{P}_0\mathrm{P}_1$	66.7%	66.3%
P_0, $\mathrm{P}_{-2}\mathrm{P}_{-1}$, $\mathrm{P}_1\mathrm{P}_2$	66.6%	66.1%

4.3.4 Combination of Parse Features

Certain parse features such as phrase in which the target word occurs and the parent phrase take on a very small number of distinct values. For example a certain target word might just occur in a noun phrase or a prepositional phrase. Thus decision trees created using just phrase of target word cannot be expected to do much better than majority classifiers. However, such features might be useful in more complicated trees involving other features. Tables 64, 65 and 66 present results obtained using decision trees which utilize multiple parse features.

Table 64: Accuracy using a combination of parse features on Senseval-2 Data.

	All	**Nouns**	**Verbs**	**Adjectives**
Majority Classifier	47.7%	51.0%	39.7%	59.0%
Head, Parent	52.6%	60.3%	40.5%	63.3%
Head, Phrase	51.9%	58.5%	40.2%	**64.0%**
Head, Parent, Phrase	52.9%	60.5%	40.9%	64.0%
Head, Parent, Phrase, Parent Phrase	52.7%	60.3%	40.6%	63.9%

Table 65: Accuracy using a combination of parse features on Senseval-1 Data.

	All	**Nouns**	**Verbs**	**Adjectives**	**Indeterminates**
Majority Classifier	56.3%	57.2%	56.9%	64.3%	43.8%
Head, Parent	65.1%	71.8%	61.1%	66.9%	60.2%
Head, Phrase	65.1%	71.5%	59.8%	**68.7%**	61.5%
Head, Parent, Phrase	65.5%	71.9%	61.7%	66.9%	61.0%
Head, Parent, Phrase, Parent Phrase	65.6%	71.8%	61.5%	67.7%	61.6%

The head-parent combination is observed to be an improvement over the accuracies achieved simply with the head word or the parent. In case of line and interest data, this improvement is significant. Amongst the

Table 66: Accuracy using a combination of parse features on *line, hard, serve* and *interest* Data.

	line	**hard**	**serve**	**interest**
Majority Classifier	54.3%	81.5%	42.2%	%
Head, Parent	60.4%	87.7%	58.1%	73.2%
Head, Phrase	54.7%	**87.8%**	45.9%	69.1%
Head, Parent, Phrase	60.4%	87.7%	57.6%	73.2%
Head, Parent, Phrase, Parent Phrase	60.5%	87.7%	56.7%	73.5%

various combinations as well, the head-parent combination has always achieved high accuracies. We note that the Head-Phrase combination has produced some of the best results for adjectives. This is significant as using only the head word or the phrase as feature has produced noticeably lower accuracies for adjectives. We also observe that the head - phrase - parent phrase combination has produced the best results for nouns.

Of all the parse feature combinations, the head-parent feature has always produced good results and thus stands out as a potent feature combination. We shall study this feature combination with other lexical and part of speech features chosen.

4.3.5 Combining Lexical and Syntactic Features

We used a simple ensemble technique to combine some of the best lexical and syntactic features identified in the previous sections. Given an instances to be disambiguated, the probability for each sense as assigned by the different decision trees is summed. The sense which gets the highest score is assigned to the instance. Tables 67, 68 and 69 show the accuracies attained on Senseval-2, Senseval-1 and *line, hard, serve* and *interest* data, respectively.

We observe a general increase in performance by combining the features. We expect better accuracies with a more complex ensemble.

Table 67: Accuracy using Combination of Features on Senseval-2 Data.

	All	Nouns	Verbs	Adjectives
Majority Classifier	47.7%	51.0%	39.7%	59.0%
Bigram, (P_0P_1)	56.0%	61.3%	48.1%	62.5%
Bigram, (P_{-1}, P_0, P_1)	56.2%	61.7%	48.2%	62.2%
Unigram, (P_0P_1)	56.7%	62.1%	49.6%	61.3%
Unigram, (P_{-1}, P_0, P_1)	57.0%	62.7%	49.8%	61.3%
Bigram, (Head, Parent)	55.9%	62.8%	45.9%	63.8%
Unigram, (Head, Parent)	56.5%	63.7%	46.9%	62.5%
(P_0P_1), (Head, Parent)	54.1%	60.7%	43.7%	63.2%
(P_{-1}, P_0, P_1), (Head, Parent)	54.3%	61.0%	43.9%	63.0%

Table 68: Accuracy using Combination of Features on Senseval-1 Data.

	All	Nouns	Verbs	Adjectives	Indeterminates
Majority Classifier	56.3%	57.2%	56.9%	64.3%	43.8%
Bigram, (P_0P_1)	69.3%	72.7%	66.9%	69.7%	68.0%
Bigram, (P_{-1}, P_0, P_1)	69.9%	73.4%	66.8%	70.0%	70.0%
Unigram, (P_0P_1)	70.3%	75.3%	67.9%	71.3%	65.8%
Unigram, (P_{-1}, P_0, P_1)	71.1%	76.4%	67.9%	71.4%	68.4%
Bigram, (Head, Parent)	69.3%	75.2%	65.7%	70.0%	65.8%
Unigram, (Head, Parent)	69.3%	75.0%	65.0%	71.0%	67.0%
(P_0P_1), (Head, Parent)	69.2%	73.6%	67.9%	66.6%	67.5%
(P_{-1}, P_0, P_1), (Head, Parent)	70.4%	76.5%	67.2%	67.4%	69.1%

Table 69: Accuracy using Combination of Features on *line, hard, serve* and *interest* Data.

	line	**hard**	**serve**	**interest**
Majority Classifier	54.3%	81.5%	42.2%	54.9%
Bigram, (P_0P_1)	71.3%	88.0%	74.6%	81.4%
Bigram, (P_{-1}, P_0, P_1)	73.1%	88.8%	76.2%	83.2%
Unigram, (P_0P_1)	68.0%	82.2%	76.6%	78.9%
Unigram, (P_{-1}, P_0, P_1)	74.2%	85.1%	81.6%	82.3%
Bigram, (Head, Parent)	72.6%	88.9%	73.3%	81.7%
Unigram, (Head, Parent)	69.3%	87.2%	75.8%	79.9%
(P_0P_1), (Head, Parent)	58.0%	86.6%	70.6%	73.7%
(P_{-1}, P_0, P_1), (Head, Parent)	62.9%	87.4%	75.1%	79.9%

4.3.6 Best Ensembles

This section summarizes the best results we have achieved for each of the Senseval-2, Senseval-1, *line, hard, serve* and *interest* data and the combinations of lexical and syntactic features that have helped achieve those results. We also indicate the combinations which we believe should yield the best results based on the *Optimal Ensemble* amongst the lexical and syntactic features. Table 70 depicts these details.

We observe that except for *line* and *hard* data, the simple ensemble achieves results better than individual classifiers. In case of *line* data, a decision tree of unigrams gives the best results, while a decision tree of bigrams performs best for *hard* data. A better ensemble technique is expected to do better than each of these as is indicated by the *Optimal Ensemble* values. The increase in accuracy is most noticeable for *serve* data which achieves 81.6% accuracy through the combination of Unigrams and part of speech features. We note that the decision tree created by the combination of the target word part of speech, and the parts of speech of its two adjacent words performs best in combination with the lexical features as opposed to the other combinations of part of speech features. The part of speech of the word to the right of the target word has been shown to be the most useful feature for sense disambiguation amongst all the individual word part of speech features. Nouns have been shown to benefit from part of speech tags on its either side while verbs and adjectives are disambiguated better using the part of speech tags of words to their immediate right. We

Table 70: The best combinations of syntactic and lexical features

Data	**Feature-Set Pair** Set 1	Accuracy	Set2	Accuracy	**Baseline Ens.**	**Majority Classifier**	**Simple Ens.**	**Optimal Ens.**
Senseval-2	Unigram	55.3%	P_{-1}, P_0, P_1	54.6%	43.6%	47.7%	57.0%	67.9%
Senseval-1	Unigram	66.9%	P_{-1}, P_0, P_1	68.0%	57.6%	56.3%	71.1%	78.0%
line	Unigram	74.5%	P_{-1}, P_0, P_1	60.4%	55.1%	54.3%	74.2%	82.0%
hard	Bigram	89.5%	Head, Parent	87.7%	86.1%	81.5%	88.9%	91.3%
serve	Unigram	73.3%	P_{-1}, P_0, P_1	73.0%	58.4%	42.2%	81.6%	89.9%
interest	Bigram	79.9%	P_{-1}, P_0, P_1	78.8%	67.6%	54.9%	83.2%	90.1%

show that the head word of a phrase is particularly useful to disambiguate adjectives. The head of the phrase and the head of the parent phrase have proved to be useful for nouns. We observe a significant amount of complementarity across lexical and syntactic features which may be exploited by a suitable ensemble technique. We have shown than guaranteed part of speech tagging, which was employed in the part of speech tagging of all the data, has helped word sense disambiguation.

5 Related Work

Extensive research has been done on word sense disambiguation during the last fifteen years. Numerous sources of information have been used for the purpose. A desire to improve accuracies has propelled the use of multiple features in combination. This gives rise to questions such as which sources of knowledge to use and how to combine the various sources of knowledge in order to produce the best results. Many different researchers have used varying techniques to combine different sets of features to achieve comparable results. However, the relative utility of the sources of information with respect to each other has not been studied in much detail. Certain lexical features such as bigrams, unigrams and surface form of the words are easy to identify in the training data. Complex features such as syntactic relations are harder to isolate since, sentences need to be parsed. It would be of interest to know if the cost of using a source of information is justified by the increase in accuracy provided. This is dependent not just on the new source of information but also on the set of features already being used.

Lexical features have been shown to attain high accuracies by Pedersen [55]. It is not clear as to how much of the accurate disambiguation done by lexical features is also done by the syntactic features or at a more finer level, how much of the disambiguation done by one feature is also done by another. This gives an idea of the redundancy of using both features. Other questions include, how much of an improvement in accuracy may one expect by using a source of information in addition to the ones already being used, also arise. This brings us to the idea of complementarity of a feature or feature set with another. Such questions have not been answered yet, however, useful insight into the issues involved and behavior of these knowledge sources, when used in combination, may be found in some of the work within the last fifteen years. Table 71 summarizes the sources of knowledge used by some of the more prominent work. It shows us that, Pedersen [55] uses just bigrams while Lee and Ng [33] and Yarowsky and Florian [76] study a comprehensive set. One of the earliest works on using multiple sources of information is that of McRoy [38]. Also, the systems of McRoy [38], Lin [35] and Stevenson and Wilks [70] disambiguate all words in the text while the rest disambiguate specific target words.

Table 71: Recent work using multiple sources of knowledge.

Author(s)	**Bi**	**Uni**	**Col**	**POS**	**Sur**	**Syn**	**D**	**Cor**	**All**
McRoy [1992]			X		X	X	X		X
Yarowsky [1995]			X					X	
Ng and Lee [1996]			X	X	X	X		X	
Lin [1997]						X		X	X
Stevenson and Wilks [1998]							X	X	X
Yarowsky [1999]			X	X	X			X	
Pedersen [2001]	X							X	
Lee and Ng [2002]	X	X	X	X		X		X	
Yarowsky and Florian [2003]	X	X	X	X	X	X		X	

Table 72: Legend for Table71

Bi	Bigrams
Uni	Unigrams
Col	Collocations
POS	Parts of Speech
Sur	Surface Form
Syn	Syntactic Features
D	Dictionary
Cor	Corpus
All	Disambiguates All Words

5.1 McRoy [1992] - TRUMP

Jacobs' TRUMP [25] [24] [26] is a system which is used for semantic interpretation, that is, given a phrase which is ambiguous, it chooses the appropriate meaning of the phrase. It does so using knowledge specific to the language in consideration. McRoy [38] incorporated a method for word sense disambiguation into TRUMP. The system aimed at disambiguating every word in the input text. She was one of the first to explore the combined use of multiple features for word sense disambiguation. This method, unlike the rest described in the section, is not corpus based, that is, there is no learning from a body of text.

TRUMP uses information about morphology, part of speech, frequency, collocations, semantic context, syntactic cues and role related expectations. Since it does not learn from a labeled corpus, TRUMP relies on an exhaustive knowledge base of a specially designed lexicon with 8,775 root words, 10,000 derivations and 13,000 senses. It consists of a core lexicon of coarse senses of a word (stem), morphological derivatives, part of speech information, syntactic constraints and frequency of usage encoded as primary or secondary senses. The coarse senses selected from the core lexicon trigger the finer senses in the dynamic lexicons. Thus, precious processing time is saved by not considering all the senses, as would have been the case if there were just one lexicon. The finer senses are considered only when relevant. There are three dynamic lexicons. The collocations lexicon lists collocations and suggested sense. It has over 1700 collocations. The domain specific lexicon has senses pertaining to specific domains. For example, the military domain has the 'attack' sense of the word 'engage'. The special abstract senses lexicon lists words with abstract senses that may at times have special meanings differing from the usual. Consider the example in the paper [38]. The word 'project' has the sense 'transfer' in the core lexicon but when used in the form below, the object is a sound and 'project' has the sense of 'a communication event'. This is encoded in the lexicon.

TRUMP's lexicon has a concept hierarchy and cluster definitions. Each sense is linked to a particular parent concept and multiple child concepts, thus forming a large semantic network. The lexicon has a thousand concepts in all. The network serves a dual purpose. Firstly, these related concepts including sibling concepts are a rich source to infer semantic context and domains. And secondly, the role related expectations can now be applied to concepts and even portions of the hierarchy instead of individual coding. A set of senses closely related to a particular concept or topic form a cluster, named after the concept. They too are used to identify semantic context. Only those senses that are not used independent of the concept, occur in the cluster. Clusters are again of three kinds. Conceptual clusters are already encoded by the concept hierarchy.

Situational clusters have senses pertaining to a certain scenario or situation. Functional clusters have senses that satisfy a particular relation. For example, part - whole relations (gun - barrel, silencer, trigger).

TRUMP automatically weights each of the features, of every sense, on a scale of -10 to 10. The sense with the highest cumulative score for all the features is chosen as the appropriate sense. If a feature is binary, that is, if it takes on one of two values, then weights corresponding to each value are chosen empirically. This weight depends on how strong a cue the particular feature is in comparison with the other cues. Features which are satisfied by a finite number of possibilities are weighted differently. Consider role related expectations. If it is known that the cue to a particular sense must be a color, the number of entries in the lexicon which satisfy the color constraint gives an idea of how strong a cue it is, if satisfied. Specifically, the strength of the cue is inversely proportional to the number of entries. This makes intuitive sense since, if all we know is that the cue is an inanimate object and this cue is satisfied, the cue will have a very small weight due to the large number of entries which satisfy it. The reciprocal of the number of elements subsumed by a concept, such as color or inanimate object, is known as specificity. The fewer the words in the concept, the stronger the probability for the word to have the corresponding sense and larger is the specificity. If a feature is not satisfied, however, the specificity is mapped to a value between 0 and -10. If a concept having low specificity is not satisfied we understand that an occurrence of high probability failed to occur, hence, providing us with a lot of information; a score close to -10 awarded. If the concept had high specificity leading us to the conclusion that something of low probability failed to occur; a negative score close to 0 assigned. Thus, specificity merits being a natural indicator of preference for a sense as compared to experimentally chosen weights. It may be noted that the concept of specificity has been used by Resnik [66] to quantify the similarity between two words. He considers the *is-a* concept hierarchy (A car is an automobile etc). Given a word, the probability that it satisfies a concept is approximated by the ratio of the number of words satisfying the concept to the total number of words. Resnik [66] defines information content as the negative logarithm of this probability, which is equal to the logarithm of its reciprocal. Thus, apart from the logarithm, which merely changes scale, information content captures the specificity of a concept.

TRUMP was used to tag a subset of the Wall Street Journal, about 2500 words. Even with the limited number of roots in its lexicon it attempted to disambiguate 91% of the words. The accuracy of the system was not evaluated due to lack of a gold standard for the corpus. There was thus no study on the utility of individual features or a comparison between them.

5.2 Yarowsky [1995]

Yarowsky [74] developed a word sense disambiguation algorithm with minimal and optionally no training data, known as co-training. The central idea behind co-training is to use a large body of unlabeled text along with a small set of labeled data to learn a classifier with reasonable accuracy. A necessary condition for co-training is two or more independent *views*. Each *view* consists of one or more features of the data which can predict the class of an instance. Thus, if there are three views, for example, then we have three sets of features which can each independently classify the instance. In co-training, a classifier is learnt using each of the views and the small set of labeled data. Yarowsky used two views. One classifier was based on the assumption that a collocation is indicative of a particular sense of the target word. This is the 'One Sense per Collocation' assumption [73]. The second classifier was based on the assumption that multiple instances of a word in a document have the same intended sense. This is the 'One Sense per Discourse' assumption proposed by Gale, Church and Yarowsky [22]. Part or all of the large unlabeled corpus is then classified using each of the classifiers individually. Only those classifications are considered for which the classifiers were confident beyond a certain pre-ascertained threshold. These instances are then added to the labeled data set and the procedure is repeated until all unlabeled data is tagged or no more instances are classified by the system. Yarowsky suggests a way to eliminate the requirement of small labeled corpus by using "seed collocations". A handful of collocations which are indicative of each of the senses of the target word are manually identified. Sentences in the unlabeled text which have these collocations are picked out and assigned the corresponding sense. These sentences act as the initial labeled data set.

Experiments were conducted using unannotated text consisting of news articles, scientific abstracts, spoken transcripts and novels. The text had around 460 million words. The words to be disambiguated were selected randomly from the words on which work has been done earlier. These include *tank, space, motion and plant* on which Schütze [67] had performed word sense disambiguation using an unsupervised algorithm. The system had an accuracy of 96.7% on words studied by Schütze who had achieved an accuracy of 92.2%. However, it should be noted that Schütze used a completely unsupervised approach. The system also performed comparably, if not better in some cases, than supervised learning methods. The final classifier based on both collocations and 'one sense per discourse' heuristic had a one percent improvement of performance than the classifier based solely on collocations (accuracies of 96.5% and 95.5%).

5.3 Ng and Lee [1996] - LEXAS

Ng and Lee [53] like McRoy were one of earliest to propose a word sense disambiguation system with large set of knowledge sources. Their implementation LEXAS is a nearest neighbor approach. The various knowledge sources utilized are listed in Table73 and described below:

Table 73: Ng and Lee Feature Sets

Notation	Description
P_{-3}, P_{-2} and P_{-1}	parts of speech of 3 words to the left
P_1, P_2 and P_3	parts of speech of 3 words to the right
M	Morphology (Singular, plural or verb form)
K1, K2, ...	co-occurring words; binary features (0, 1)
C1, C2, ..., C9	9 Collocations; binary features (0, 1)
V1, V2, ...	verbs in verb object relation with target word

- **Part of Speech:** The features L_1, L_2 and L_3 have the form L_w where L_w takes the part of speech of the word **w** positions to the left of the target word, as its value. Similarly, the features R_1, R_2 and R_3 take on the part of speech of the words to the right of the target word.

- **Morphology:** *M* stands for the morphological form of the target word. *M* takes on a value depending on the part of speech of the word. If the word is a noun, *M* is either singular or plural. In case of verbs, *M* is either the infinitive (for eg. in case of eat), present-participle(eg. eats), past(eg. ate), present-participle(eg. eating) or past-participle(eg. eaten).

- **Co-occurrences:** K1, ..., Km correspond to the target word's *m* co-occurring words. This set of co-occurring words is elected for each of the target words based on conditional probability. All words in the training corpus are considered as probable co-occurrences. A count (Nc) is made of the number of times a candidate co-occurrence (Kc say) occurs in the training text and in those instances a count (Nc,i) of the number of times the target word has a certain sense i. Kc is chosen as a co-occurrence only if the ratio of Nc,i and Nc is greater than a certain pre-ascertained threshold. Ng and Lee chose a threshold of .8. Additionally, Kc must occur at least 5 times in the training corpus such that the target word has sense i. At most 5 co-occurring words were chosen for every sense of the target word. Thus the number of co-occurrence features is at most 5 x n, if n is the total number of senses. Given a test instance, these features take on a value of 1 or 0 depending on the presence or absence of these co-occurring words in the test sentence.

- **Collocations:** C1, ..., C9 are collocations involving the target word. They too are extracted by a similar application of conditional probability, picking those that are suggestive of a particular sense. Collocations differ from occurrences primarily as they prescribe a fixed sequence of the words. For example, *interest rate* is a collocation as *rate* is present immediately after *interest*. The phrase *rate your interest* does not have the collocation *interest rate*. For a given sentence, Ng and Lee considered 9 possible sequence of words as candidate collocations. This set of 9 collocations is summarized by the table in their paper [53] reproduced below in table 74. These features too, like co-occurrences are binary. Given a test sentence, the features take on values 1 or 0 depending on presence or absence of appropriate collocation.

- **Verbs:** Verbs (V1,...) in a verb object relation with the target word and indicative of a particular sense of the noun being disambiguated. Selection of verbs is again based on conditional probability.

LEXAS constructs an exemplar-based classifier for each target word. The training examples after conversion to corresponding feature vectors are stored. Given a test example, the corresponding feature vector is compared with the feature vectors of the training examples. Ng and Lee implemented the nearest neighbor and not the k-nearest algorithm, thus, the sense of the training example closest to the test set was chosen as the intended sense of the word, in the context.

The system was evaluated on the *interest* [13] data. 100 experiments were conducted using randomly sampled 600 sentences as test data and the remaining 1769 as training data. The authors achieved an accuracy of 87.4% with a standard deviation of 1.37%. The authors studied the contribution of individual sources of information by conducting experiments with individual sources of information. They found that collocations were most accurate (80.2%) followed by part of speech and morphological form (77.2%). Co-occurrences (62.0%) and verb object relations (43.5%) were found not to be as helpful. The authors, who had used a one

Table 74: Candidate collocation Sequences.

Notation	Left Offset	Right Offset	Collocation Example
$C_{-1,-1}$	-1	-1	accrued interest
$C_{1,1}$	1	1	interest rate
$C_{-2,-1}$	-2	-1	principal and interest
$C_{1,2}$	1	2	interest and dividends
$C_{-3,-1}$	-3	-1	sale of an interest
$C_{-2,1}$	-2	1	in the interest of
$C_{-1,2}$	-1	2	an interest in a
$C_{1,3}$	1	3	interest on the bonds

sentence window for co-occurrences, believe that a bigger window will help. This, however, is the extent of the study of the features. No study was made on the redundancy and complementarity of the features.

5.4 Lin [1997]

Lin's [35] developed a supervised approach to word sense disambiguation which did not require a classifier for every word to be disambiguated. He proposed that, since, the different senses of a word are more or less used in disparate contexts, the sense of a word is usually similar to the sense of a *different* word having a similar context. Given a test instance, the system identifies the local context of the target word. It then identifies other words in the training corpus which have similar contexts. The sense of the target word which is most similar to these words is chosen as the desired sense.

This approach eliminates the need of a sense tagged corpus for every word to be disambiguated. A smaller training corpus is used like Yarowsky [74], however, the basic methodology of disambiguation is very different. The tradeoff of a small training corpus is the lack of modeling specific for every word, thereby affecting accuracy. Another distinction of Lin's [35] methodology is that the system aims to disambiguate all the words in the input text as compared to just certain target words given their context.

Lin [35] relies completely on syntactic dependencies to capture the context of a word. Syntactic dependencies are grammatical relations such as subject-verb, adjunct-verb and verb-object relations shared by two words in a sentence. It may be noted that adjuncts are modifiers of the verb which are not necessary for the validity of the sentence. For example:

$$\texttt{Damienplayedtheguitaradmirably}. \tag{27}$$

In the above sentence, *Damien* and *played* are related by the subject-verb relation, *played* and *guitar* form the verb - object pair, while, *played* and *admirably* share the verb - adjunct relation. Without the adjunct the sentence is still grammatically valid. It may also be noted that in all such word pairs, one of them is known as the head while the other as the modifier. For example, the verbs are modified by their objects and adjuncts. The author captures the context of a word with sets of the above mentioned relations. Each set constitutes - the type of syntactic dependency the word is involved in, the word with which it shares the relation and whether the target word is the head or modifier. Associated with each such set is its suggestiveness to a particular sense. This is captured by another trio - the word related to the target word via the syntactic relation, the number of times the word was found in the same context and the log likelihood ratio indicating the suggestiveness of a particular sense, calculated following Dunning [19].

A broad coverage parser is used to parse the text. A database of local contexts(lc) and word-frequency-likelihood(C(lc)) information is extracted from the training corpus. The Walls Street Journal corpus was used for this purpose. The system was evaluated on a subset of SemCor [39], the "press reportage" part. SemCor is text taken from the Brown corpus. All the nouns, verbs, adjectives and adverbs in it are tagged with senses from the WordNet. This is primarily because SemCor is the only significant repository of text with all words sense tagged. Lin believes that the subtle distinctions in senses are hard for even a human to differentiate and hence used allowed an assigned sense to be correct if it was close enough to the correct sense. With varying levels of strictness the system achieved accuracies between 59% and 67%. The contributions of individual dependency relations to the final disambiguation were not studied.

5.5 Wilks and Stevenson [1998]

Wilks and Stevenson [70] like McRoy [38] and Lin [35] developed a method for word sense disambiguation on all words of a text. They too use a dictionary. But unlike McRoy [38] they use a training corpus as well and thus have a empirical learning approach. They use the Longman Dictionary of Contemporary English

(LDOCE) [59] to assign senses. They use a set of filters and partial taggers. The filters weed out senses which do not agree with certain sources of information, while the partial taggers give a set of probable senses based on certain other sources of information. Following is a list of filters and partial taggers used:

- **Part of Speech Filter:** The first feature to be utilized is the part of speech. Only those senses corresponding to the part of speech assigned by the tagger merit future consideration. Brill Tagger [8] was used to tag the words.
- **Dictionary Definitions Partial Tagger:** Semantic context is captured to some extent by a modified version of the simulated annealing proposed by Cowie [17]. Lesk [34] had originally proposed a way to disambiguate the sense of a word by choosing that sense whose definition has the maximum number of words in common with that of the definitions of other words in the sentence. If A, B, C and D are a sequence of words in a sentence and the senses of A, B and D are known, the definitions of the correct senses of A, B and D may be used to determine the correct sense of C. However, if we are to determine the correct sense of all the words, we have to start with a guess of the sense of all the words. The sense of a word (B say) may be changed based on the definitions of the others. Then, the sense of C may be changed, in a similar fashion. This raises a doubt on the correct sense of B as it was calculated based on the wrong sense of C. Cowie [17] came up with a numerical method to solve this problem which comes under the general class of simulated annealing. Wilks and Stevenson propose a further modification of this method, wherein, the method gave out a set of most probable senses for each word and not just one. Additionally, the original approach had a bias to senses with longer dictionary definitions. The method was thus modified such that the number of matches was divided by the total number of words in the definition, thereby, eliminating the bias.
- **Pragmatic Codes Partial Tagger:** LDOCE is rich in rules which help determine the broad subject pertaining to a sense. These rules, known as pragmatic codes, are used to disambiguate nouns. Senses of nouns have been shown to be sensitive to the topic being discussed (Gale [22]). Once again a modified form of simulated annealing algorithm is used so that the number of pragmatic codes indicating a certain subject area is maximized.
- **Selectional Restriction Partial Tagger:** Selectional restrictions encoded in LDOCE capture role related expectations. LDOCE has rules for every content word listed in it. Thirty-five semantic

classes of nouns such as ‘Human’, ‘Solid’ and ‘Plant’, defined in it, are assigned to each sense of all the nouns, verbs, adjectives and adverbs. A verb is assigned the classes of its expected subject, object and indirect object. Adjectives - the noun they modify and Adverbs - their modifier. Syntactic relations amongst the words in the input are identified via the syntactic analyzer(Stevenson [68]). Those senses are chosen which uphold the restrictions encoded.

Like Yarowsky [74] a decision list is used as the classifier. It has an ordered set of rules along with an associated sense. The sense corresponding to the first rule satisfied by an instance is chosen as the desired sense of the target word in that instance. A supervised learning algorithm learns an appropriate decision list based on the results of the partial taggers, the correct sense, the frequency distribution of various senses and a set of 10 collocations. The collocations chosen comprise the first noun, verb, and preposition to the left and right of the word being disambiguated. It also consists of the first and second words to the left and right of the target word. Given a test sentence, the taggers and filters are applied on it. The results along with the frequency information and the collocations set is then given to the decision list which does the classification. The authors believe that even though the decision list is trained on instances corresponding to a few words, it may be used to classify any word in LDOCE.

The system was tested on a subset of SemCor [39]. It may be noted that Wilks and Stevenson [70] and Lin [35] who also had an all words disambiguation system evaluated it on a subset of SemCor as well. As the system was based on the LDOCE senses, the mapping between the LDOCE and WordNet senses done by Bruce and Guthrie [12] was used. The authors found that they received accuracies of around 55-58% when the dictionary definitions, pragmatic codes and selectional restrictions were used individually. When all the sources of information were used together they achieved an accuracy of 59%. The authors believe that the increase encourages the use of multiple sources of information. The dictionary definitions were most useful for disambiguation, however, selectional restrictions and pragmatic codes, in that order, were not far behind. The authors do not give details on the redundancy and complementarity of the knowledge sources.

5.6 Yarowsky [1999]

Yarowsky [75] describes a word sense disambiguation system using hierarchical decision lists and a rich set of features. Hierarchical decision lists allow modeling specific to certain kinds of instances. For example, we could want to model the classifier differently for different parts of speech or surface forms. The hierarchy is grown and validated on the training data itself to check if the addition has improved accuracy. If not the addition is undone. At each node, the training data is split and all learning in this path is based on this subset of the data. This is similar to decision trees, however, the number of nodes is much smaller. Thus although, there is some modeling specific to certain key features, training data is not split as much as in decision trees.

Yarowsky [75] uses a rich set of features which includes a combination of positional options and word information listed in Table75. The positional option lists the position of the word relative to the target word, whose information will be used. The positions considered are are described below:

- **Relative Offset:** This corresponds to words which are close to the target word. A relative offset of **1** signifies the information of the word on the immediate right of the target word will be utilized. Similarly, **-2** corresponds to information of word at two positions to the left of the target word. **-2, -1** and **1** are the possible values considered.
- **Syntactically related words:** This corresponds to words which are syntactically related to the target word. For example, the words which share subject–verb or verb–object relation with the target word.
- **Co-occurrences:** Words within a window of $\pm$ k words around the target word, where k is a pre-ascertained constant.
- **Collocations:** Word sequences corresponding to $C_{1,2}$ and $C_{-1,1}$ as per the notation described earlier. It may be noted that the former sequence, corresponding to the word to one position to the right and a word two positions to the right of the target word, was used by Ng and Lee [53]. $C_{-1,1}$ corresponding to the word to the left and right of the target word was not used by them.

The information used of words at these positions is the *Word Information*. The kinds of *Word Information* used are described below:

- Surface form of the word, for example whether the word is *eat, eats, ate* or *eaten*.

- Root of the word, for example the root word of the words mentioned above is *eat*.
- Part of speech of word, such as proper noun and verb in past tense.
- Answer to certain questions, such as, if the target word is capitalized.

Table 75: Positional Options and Word Information

Positional Options	Word Information
Relative offset (1, -1, -2)	Literal (surface form)
Target word (0)	Lemma (stem)
Collocations	Part of Speech
Co-occurrences($\pm$k word window)	Question (e.g. is word capitalized)
Syntactic relations	

Due to the many combinations possible between the positional options and word information, only those features are selected which are indicative of a particular sense. The conditional probability criterion as mentioned earlier is employed for the purpose. The system took part in the Senseval-1 exercise held in the summer of 1998 and achieved and accuracy of 78.4%. No analysis of the contribution of individual knowledge sources was made.

5.7 Pedersen [2001]

Pedersen [55] describes a system for word sense disambiguation using just bigrams as source of information. He conducts experiments with decision trees, decision stump, bayesian classifier and a majority classifier. A decision stump is a one node decision tree. The feature which best disambiguates the training data is selected for the node. Power divergence statistic [18] and the Dice Coefficient were both used independently to select the appropriate bigrams. A hundred bigrams each were selected for each task based on power divergence and the Dice Coefficient. There must be at least five instances of these bigrams in the training data.

Senseval-1 data was used for training and evaluation. A decision tree along with the power divergence statistic gave the best results with accuracies of 19 out of the total 36 tasks being above the best results in

Senseval-1. Pedersen believes that bigrams with decision trees can be very accurate in predicting the correct sense of a word. The significance of this result lies in the fact that bigrams, which are a kind of lexical feature, are easy to capture and yet very powerful in disambiguation. Thus, decision trees of bigrams act as a powerful baseline to build on and thus the added gravity in determining the worth of utilizing more complex sources of information. He believes that in case of word sense disambiguation, identifying good features to use for disambiguation is vital. A good feature will get good results with a number of learning algorithms, but the converse is not true. Pedersen also points out that decision trees help understand the relations between the features and are thus a good choice to use as they will in turn help identify features that are most useful in discriminating the senses of a words.

5.8 Lee and Ng [2002]

Lee and Ng [33] have performed experiments with a number of knowledge sources and supervised learning algorithms (listed in Table76). The study is probably the first which takes a comprehensive look at the knowledge sources, algorithms and the interaction between them.

Table 76: Lee and Ng - Sources of Knowledge and Supervised Learning Algorithms

Sources of Knowledge	**Supervised Learning Algorithms**
Part of speech of neighboring words	Support Vector Machines
Unigrams	Naive Bayes
Local Collocations	AdaBoost
Syntactic Relations	Decision Trees

Since this thesis is focussed on knowledge sources, the behavior of the various algorithms is beyond its scope. The sources of knowledge studied are the following:

- Part of speech of three words to the left and right of the target word along with that of the target word. Ratnaparkhi [65] is used to part of speech tag the data.
- Unigrams that are indicative of a particular sense of the target word They are selected based on the

conditional probability of a sense given the unigram, as discussed earlier.

- Word sequences occurring at pre-decided positions relative to the target word are considered as candidate collocations. The eleven pre-decided positions are given below:

  $C_{-1,-1}$, $C_{1,1}$, $C_{-2,-2}$, $C_{2,2}$, $C_{-2,-1}$, $C_{-1,1}$, $C_{1,2}$, $C_{-3,-1}$, $C_{-2,1}$, $C_{-1,2}$, $C_{1,3}$

  Given $C_{i,j}$ C stands for collocation and consists of tokens from position i to position j, relative to the target word. It may be noted that these subsume the 9 collocational features used by Ng and Lee [53]. The two additions are $C_{-2,-2}$ and $C_{2,2}$. These two are unique since they do not include the word adjacent to the target word.

- The Charniak parser [14] is used to parse the instances enabling the use of various syntactic relations as features. The relations used depend on the part of speech of the target word. If a noun, the following features are used: head word of parent phrase, its part of speech, its voice (active or passive) and its relative position from the target word. In case of verbs, the following six features are used: a word closest to the target word on its left which has the target word as parent, its part of speech, similar features based on a word to the right, the part of speech and voice of the target word. If the word to be disambiguated is an adjective, just two features are used - the head word of its parent phrase along with its part of speech.

The experiments were done on Senseval-1 and Senseval-2 [20] data. As mentioned earlier standard test and training texts occur for these data. The best results were achieved using all knowledge sources and support vector machine. Since, this thesis is on decision trees all further discussion of results will be on experiments conducted with them. All knowledge sources out perform the best even in case of decision trees when run on Senseval-1 data (accuracy of 73.4%). On Senseval-2 data, the system using collocations (accuracy of 57.2%) as the knowledge source is found to be best. In fact, part of speech tags (accuracy of 55.3%) and syntactic relations (54.2%) also do better than the combined classifier. The unigrams were found to be the weakest sources of knowledge when run on both Senseval-1 (accuracy of 66.2%) and Senseval-2 (accuracy of 50.9%) data. Based on their experiments, the authors conclude that no knowledge source alone encompasses the knowledge provided by other. On the contrary, a combination of the many sources has yielded best results.

5.9 Yarowsky and Florian [2002]

Yarowsky and Florian [76] have done experiments on word sense disambiguation using six supervised learning algorithms and variations of the data representation. That is, apart from using various knowledge sources, they conducted experiments with varying context size, size of training data and number of senses distinctions per task. The knowledge sources and algorithms studied are listed in Table77. The authors classify the algorithms into two categories. The decision list and transformation based learning model (TBL) are discriminative algorithms which rely on a small subset of features for disambiguation while the rest are agglomerative classifiers which base their decision on a weighted sum of all the knowledge sources.

Table 77: Yarowsky and Florian - Sources of Knowledge and Supervised Learning Algorithms

Sources of Knowledge	**Supervised Learning Algorithms**
Bag of words	Three kinds of Bayes Classifier
Collocations	Decision List
Syntactic Relations	Transformation based learning model (TBL)

The knowledge sources used are classified into three categories. One, bag of words, which includes morphological information such as the lemma, apart from the surface form of the words. Two, local context captured by collocations, which includes bigrams and trigrams within a certain window around the target word. And three, syntactic features such as verb object relationship and subject verb relations.

Experiments were conducted on Senseval-2 data. The experiments reveal that on dropping any of the three categories of knowledge sources, there is a drop in accuracy of the system for all the algorithms considered. Yarowsky and Florian note that the drop is significant for aggregative classifiers and not for the discriminative classifiers which is in support of their theory that discriminative classifiers make their decision based on a small set of features. Decision list for example, will classify based on a single most suggestive feature. The drop in performance of the decision lists with any of the three kinds of information was the most uniform (2.3% - 4.5%). It may be noted, that although decision trees were not studied in this paper, they too are a kind of discriminative classifier, who make decisions based on a small subset of features. Of course, not as much so as the decision lists.

The drop in performance when the bag of words knowledge source was left out showed a lot of variance with the different algorithms. While TBL was almost not affected, a drop of only 0.5%, the Bayes had a drop of 15%. On an average, collocations boosted the performance by 3.3%. As discovered by Ng and Lee [53], syntactic features were again found to provide only marginal improvement in performance (1.4%). Additionally, Yarowsky and Florian discover that there is only a little improvement in disambiguation of nouns with syntactic features if collocations are being used already. However, verbs and adjectives show significant improvement with syntactic features. Yarowsky and Florian, like Lee and Ng [33] conclude that a single knowledge source alone does not provide the best results and a suitable combination would be ideal.

5.10 Pedersen [2002]

Pedersen [56] did a pairwise study of the various systems which participated in the Senseval-2 exercise to disambiguate the English and Spanish tasks. The study primarily looks as the systems as black boxes and does not delve into the algorithms involved or the knowledge sources used. Pedersen [56] introduces the term *optimal combination* which is defined as follows:

> Optimal combination is the accuracy that could be attained by a hypothetical tool called an optimal combiner that accepts as input the sense assignments for a test instance as generated by several different systems. It is able to select the correct sense from these inputs, and will only be wrong when none of the sense assignments is correct.

Pedersen states that albeit hypothetical, the optimal combination provides an upper bound to the accuracy that may be attained by combining multiple systems. In order to determine the optimum combination of a pair of systems, the test instances were put one of four categories - correctly classified by both systems, correctly classified by system one but not by two, correctly classified by system two and not one and incorrectly classified by both systems. The optimum combination is then determined by the ratio of number of instances in the first three categories to the total number of instances. Pedersen also studies the similarity between each pair of systems, that is, how often is the classification of system one, the same as that of system two, irrespective of whether the classification is correct or not. The Kappa statistic [15] is used for this purpose. He believes that those pairs which have low similarity and high optimal combination are of interest as these pairs are getting different kinds of instances right and if combined appropriately may yield a system with high accuracy. He points out that such systems are complimentary as in a number of instances which are tagged incorrectly a system are tagged correctly by the other. Of course, systems with high similarity and high optimal combination provide high accuracy without combination itself. The author points out that such systems may have a high optimal combination but they are also highly redundant as in a large number of instances are tagged identically by the two systems. Combining systems will be suitable only if the gain in accuracy is worth the effort.

All the research above considered, there still remain questions regarding the use of multiple sources of knowledge for word sense disambiguation. The Lee and Ng [33] and the Yarowsky and Florian [76] papers especially show the benefit of multiple knowledge sources but do not shed light on the upper bound of the accuracy of a system using multiple sources. Pedersen [56] describes a method to determine an upper bound on the accuracy achievable by a set of systems. This thesis aims at extending this idea to determine an upper bound in accuracy achievable by one system by combining multiple sources of information. Given two sources of knowledge - A and B say, a classifier based on each alone may achieve a certain number of correct classifications. What is of interest is how many instances correctly classified using A are also correctly classified using B. This gives an idea of the redundancy in the information attained from knowledge sources A and B. On the other hand, the number of instances correctly classified using A which were wrongly classified on using B, gives an idea of the amount A complements the information attained by B alone. Similarly, also of interest is the amount B complements A. This idea of redundancy and complementarity of knowledge sources is an instantiation of the redundancy and complementarity of systems described in Pedersen [56]. Knowledge sources apart, systems may be redundant or compliments due to their algorithm as well. These are some of the questions this thesis hopes to address regarding syntactic cues and lexical features.

6 CONCLUSIONS

A rich set of features may be used to represent written text. This thesis suggests that lexical and syntactic features are both useful for word sense disambiguation. It is expected that a number of instances which are correctly tagged using lexical features are also correctly disambiguated using syntactic features. However, this thesis takes the view that there are a significant number of instances which are tagged correctly just by the lexical features or the syntactic features alone. A suitable ensemble technique may be used to combine the lexical and syntactic features to benefit from this complementarity and attain higher accuracies than individual features. We use the Senseval-2, Senseval-1, *line, hard, serve* and *interest* data which together consist of around 50,000 sense tagged instances and almost all the sense tagged text available in the research community. In order to utilize syntactic features for word sense disambiguation, we developed a package `posSenseval` [48] to part of speech tag the data using the Brill Tagger [8] [9] [10]. We identified a limitation in the Brill Tagger and corrected it by a technique known as *Guaranteed Pre-Tagging* described by Mohammad and Pedersen [42]. We developed the package `parseSenseval` [47] to parse the data using the Collins Parser. We identified and documented spurious instances which did not conform to the data format and provide a cleaned up version of the data.

We conducted an extensive array of disambiguation experiments on the part of speech tagged and parsed data utilizing lexical and syntactic features. We found both lexical and syntactic features produced reasonably good accuracies when used individually. We show that the part of speech of the word to the right of the target word is particularly useful for sense disambiguation as compared to all the other individual word part of speech features. We show that nouns benefit from the part of speech of words on its either side while verbs and adjectives gain maximum from the part of speech of words immediately to the right. A combination of individual part of speech tags attains even better accuracies and we identify the combinations (P_0, P_1) and (P_{-1}, P_0, P_1) as the most potent part of speech combinations. By potent we mean that these combinations attain very high accuracies for all data and albeit, combinations formed by many more part of speech features might at times yield better results, the improvement is not significant. We show that the head word of a phrase (head for short) is particularly useful to disambiguate adjectives. The head and the head word of the parent phrase (parent for short) have been useful for nouns. We identify the head and parent combination as the most potent parse feature combination. This combination, as compared to the other parse feature combinations, has been shown to give consistently high accuracies for all the data.

We introduce the measures of *Baseline Ensemble* and *Optimal Ensemble* to quantify the redundancy and complementarity amongst two separate feature sets. We found that the syntactic and lexical features showed a considerable amount of complementarity. This suggests that a suitable ensemble may indeed produce results significantly better than a system based on just lexical or syntactic features. We conducted experiments using a very simple ensemble technique to show that we do indeed get an increase in accuracy by combining lexical and syntactic features. The discriminating knowledge provided by the part of speech of the word immediately following the target word (P_1) is shown to be less similar to lexical features than other relevant individual part of speech features. This along with the high optimal combination values with the lexical features suggests that P_1 is significantly complementary to the lexical features as compared to other individual part of speech features. We also show that the optimal combination is in general lower within syntactic features and lexical features as compared to across syntactic and lexical features. This again suggests that there is significant benefit to combining lexical and syntactic features. We show that the decision tree created by the combination of the target word part of speech, and the parts of speech of its two adjacent words performs best in combination with the lexical features as opposed to the other combinations of syntactic and lexical features.

By comparing the results of sense disambiguation using data which was part of speech tagged with and without *Guaranteed Pre-Tagging* [42], we show that the former has improved word sense disambiguation.

7 Future Work

In this thesis we used a very simple ensemble technique to combine lexical and syntactic features, which achieved reasonable accuracies. We also showed the optimal ensemble which acts as an upper bound to the accuracy achieved on combining two sets of features. A question that merits consideration is choosing between an ensemble of decision trees based on lexical and syntactic features versus one decision tree created by a combination of lexical and syntactic features. The former has the advantage of having separate trees based on different kinds of features enabling us to draw conclusions about the interactions amongst a particular kind of features which suggests the intended sense. However, creating one tree will yield higher accuracies if lexical and syntactic features share a relation as described below.

Consider a lexical feature L and a syntactic feature S such that L = 1 AND S = 1 is strongly suggestive of a sense X. Two individual trees created based on L and S separately may not be good at disambiguation and thus the ensemble itself is not expected to do much better. However, a decision tree created based on the combination of features L and S is expected to correctly identifying X as the intended sense. At present it remains unclear if lexical and syntactic relations share such relations which suggest a combination of features in a single decision tree.

This thesis demonstrates the effects of certain lexical and syntactic features on test instances in general. However, we have identified certain features to be particularly useful for target words of certain parts of speech while not so much for other. For example, the head word of the phrase is particularly suited for adjectives and the head word of the parent phrase has been shown to be useful in disambiguation of only the nouns. Part of speech features of words to the right of the target word achieved higher accuracies for verbs and adjectives while nouns were helped by part of speech features on either side. We believe that the overall accuracy of the system may be improved if we use a different optimal set of features is used to disambiguate words of different parts of speech.

This thesis uses decision trees for word sense disambiguation but we believe that the redundancy and complementarity of lexical and syntactic features is independent of the learning algorithm. Unigrams, due to the large number of features involved, may be better suited for Naive Bayesian classifiers. Part of speech and parse features may be more suited for decision trees and other machine learning algorithms that build representation models of the training data. It will be of interest to investigate the redundancy and complementarity of feature sets across a range of supervised learning algorithms. If the complementarity varies, it will mean that different learning algorithms capture differing discriminating knowledge from the same features and thus impact the complementarity and redundancy across two sets of features.

References

[1] S. Abney. Partial parsing via finite state cascades. In *Proceedings of the ESSLLI'96 Robust Parsing Workshop*, Prague, Czech Rep., 1996.

[2] S. Abney. Part-of-speech tagging and partial parsing. In *Corpus-Based Methods in Language and Speech Processing*, pages 118–136, Dordrecht. Netherlands, 1997. Kluwer Academic Publishers.

[3] S. Banerjee and T. Pedersen. Sval1to2, v-0.31, 2001. www.d.umn.edu/˜tpederse/data.html.

[4] S. Banerjee and T. Pedersen. N-gram statistics package, v-0.59, 2003. www.d.umn.edu/˜tpederse/nsp.html.

[5] S. Banerjee and T. Pedersen. Sensetools, v-0.3, 2003. www.d.umn.edu/˜tpederse/sensetools.html.

[6] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[7] L. Breiman, J. Friedman, and C. Stone. *Classification and Regression Trees*. Wadsworth International, 1984.

[8] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Computational Linguistics*, Trento, Italy, 1992.

[9] E. Brill. Transformation-based error-driven parsing. In *Proceedings of the 3th International Workshop on Parser Techn ologies*, Tilburg, The Netherlands, 1993.

[10] E. Brill. Some advances in rule-based part of speech tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA, 1994.

[11] E. Brill and R. Mooney. An overview of empirical natural language processing. *AI Magazine*, 18(4):13–24, 1997.

[12] R. Bruce and L. Guthrie. Genus disambiguation: A study in weighted preference. In *Proceedings of International Conference of Computational Linguistics (COLING)*, pages 1187–1191, Nantes, France, 1992.

[13] R. Bruce and J. Wiebe. Word-sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 139–146, 1994.

[14] E. Charniak. A maximum-entropy-inspired parser. *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132–139, 1999.

[15] J. Cohen. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.*, 20:37–46, 1960.

[16] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing.* Ph.D. dissertation, University of University of Pennsylvania, Philadelphia, 1999.

[17] J. Cowie, J. A. Guthrie, and L. Guthrie. Lexical disambiguation using simulated annealing. In *Proceedings of the International Conference on Computational Linguistics*, pages 359–365, 1992.

[18] N. Cressie and T. Read. Multinomial goodness of fit tests. *Journal of the Royal Statistics Society Series B*, 46:440–464, 1984.

[19] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[20] P. Edmonds and S. Cotton, editors. *Proceedings of the Senseval–2 Workshop*. Association for Computational Linguistics, Toulouse, France, 2001.

[21] R Florian and D Yarowsky. Modeling consensus: Classifier combination for word sense disambiguation. In *Proceedings of EMNLP'02*, pages 25–32, 2002.

[22] W. Gale, K. Church, and D. Yarowsky. One sense per discourse. In *Proceedings of the Fourth DARPA Speech and Natural Language Workshop*, 1992.

[23] G. Holmes, A. Donkin, and I.H. Witten. Weka: a machine learning workbench. In *Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems*, pages 357–361A, Brisbane, Australia, 1994.

[24] P. Jacobs. A knowledge framework for natural language analysis. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI-87)*, Milan, Italy, 1987.

[25] P. Jacobs. Language analysis in not-so-limited domains. In *Proceedings of the Fall Joint Computer Conference.*, Dallas, TX, 1987.

[26] P. Jacobs. Trump: A transportable language understanding program. In *Technical Report CRD89/181, GE Research and Development Center.*, Shenectady, NY, 1989.

[27] A. Kilgarriff. What is word sense disambiguation good for? In *Proceedings of the NLP Pacific Rim Symposium*, Phuket, Tailand, 1997.

[28] A. Kilgarriff. Senseval: An exercise in evaluating word sense disambiguation programs. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 581–588, Granada, Spain, 1998.

[29] A. Kilgarriff. Special double issue on senseval. In *Computers and the Humanities*, number 33, pages 4–5, 2000.

[30] D. Klein, K. Toutanova, T. Ilhan, Kamvar S., and C. Manning. Combining heterogeneous classifiers for word-sense disambiguation. In *ACL-2002 Workshop on Word Sense Disambiguation*, 2002.

[31] C. Leacock, M. Chodorow, and G. Miller. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165, March 1998.

[32] C. Leacock, G. Towell, and E. Voorhees. Corpus-based statistical sense resolution. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 260–265, March 1993.

[33] K.L. Lee and H.T. Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 41–48, 2002.

[34] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In *Proceedings of SIGDOC '86*, 1986.

[35] D. Lin. Using syntactic dependency as a local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 64–71, Madrid, 1997.

[36] D. Lin. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*, Granada, Spain, May, 1998.

[37] Collins M. Discriminative reranking for natural language parsing. In *Proceedings of the 17th International Conference on Machine Learning*, pages 175–182. Morgan Kaufmann, San Francisco, CA, 2000.

[38] S. McRoy. Using multiple knowledge sources for word sense discrimination. *Computational Linguistics*, 18(1):1–30, 1992.

[39] G. Miller, M. Chodorow, S. Landes, C. Leacock, and R. Thomas. Using a semantic concordance for sense identi cation. In *Proceedings of the ARPA Human Language Technology Workshop.*, pages 240–243, 1994.

[40] T. Mitchell. *Machine Learning*. McGraw–Hill, Boston, MA, 1997.

[41] S. Mohammad and T. Pedersen. Brillpatch, v-0.1, 2002. www.d.umn.edu/˜tpederse/pos.html.

[42] S. Mohammad and T. Pedersen. Guaranteed pre-tagging for the brill tagger. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 148–157, Mexico City, 2002.

[43] S. Mohammad and T. Pedersen. Senseval1-fix, v-0.1, 2002. www.d.umn.edu/˜tpederse/data.html.

[44] S. Mohammad and T. Pedersen. hardonetwo, v-0.1, 2003. www.d.umn.edu/˜tpederse/data.html.

[45] S. Mohammad and T. Pedersen. interestonetwo, v-0.1, 2003. www.d.umn.edu/˜tpederse/data.html.

[46] S. Mohammad and T. Pedersen. lineonetwo, v-0.1, 2003. www.d.umn.edu/˜tpederse/data.html.

[47] S. Mohammad and T. Pedersen. parsesenseval, v-0.1, 2003. www.d.umn.edu/˜tpederse/data.html.

[48] S. Mohammad and T. Pedersen. possenseval, v-0.1, 2003. www.d.umn.edu/˜tpederse/data.html.

[49] S. Mohammad and T. Pedersen. refine, v-0.1, 2003. www.d.umn.edu/˜tpederse/data.html.

[50] S. Mohammad and T. Pedersen. serveonetwo, v-0.1, 2003. www.d.umn.edu/˜tpederse/data.html.

[51] S. Mohammad and T. Pedersen. Sval2check, v-0.1, 2003. www.d.umn.edu/˜tpederse/data.html.

[52] S. Mohammad and T. Pedersen. unique-hard, v-0.1, 2003. www.d.umn.edu/˜tpederse/data.html.

[53] H.T. Ng and H.B. Lee. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 40–47, 1996.

[54] T. Pedersen. A simple approach to building ensembles of Naive Bayesian classifiers for word sense disambiguation. In *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 63–69, Seattle, WA, 2000.

[55] T. Pedersen. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 79–86, Pittsburgh, 2001.

[56] T. Pedersen. Assessing system agreement and instance difficulty in the lexical samples tasks of senseval-2. In *Proceedings of the ACL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 40–46, Philadelphia, 2002.

[57] T. Pedersen. Evaluating the effectiveness of ensembles of decision trees in disambiguating senseval lexical samples. In *Proceedings of the ACL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 81–87, Philadelphia, 2002.

[58] T. Pedersen and R. Bruce. Knowledge lean word sense disambiguation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 800–805, Madison, WI, 1998.

[59] P. Procter, editor. *Longman Dictionary of Contemporary English*. Longman Group Ltd., Essex, UK, 1978.

[60] A. Purandare and T. Pedersen. Senseclusters, v-0.37, 2003. www.d.umn.edu/˜tpederse/senseclusters.html.

[61] J. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[62] J. Quinlan. Rule induction with statistical data - a comparison with multiple regression. *Operational Research Society*, 38:347–352, 1987.

[63] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1992.

[64] R. Quinlan. Bagging, boosting, and c4.5. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 725–730, Menlo Park, CA, 1996.

[65] A. Ratnaparkhi. A maximum entropy part-of-speech tagger. *In Proceedings of the Empirical Methods in Natural Language Processing Conference*, pages 133–142, 1996.

[66] P Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 448–453, 1995.

[67] H. Schütze. Dimensions of meaning. In *Proceedings of Supercomputing '92*, pages 787–796, Minneapolis, MN, 1992.

[68] M. Stevenson. Extracting syntactic relations using heuristics. In *Proceedings of the European Summer Schoolon Logic, Language and Information.*, Saarbrucken, Germany, 1998.

[69] D. Tufis and O. Masin. Tagging romanian texts: a case study for qtag, a language independent probabilistic tagger. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, pages 589–596, Granada, Spain, 1998.

[70] Y. Wilks and M. Stevenson. Word sense disambiguation using optimised combinations of knowledge sources. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics*, 1998.

[71] T. Winograd. *Understanding Natural Language*. Academic Press, Inc., 1972.

[72] I. Witten and E. Frank. *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan–Kaufmann, San Francisco, CA, 2000.

[73] D. Yarowsky. One sense per collocation. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 266–271, 1993.

[74] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, MA, 1995.

[75] D. Yarowsky. Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities*, 34(1–2):179–186, 1999.

[76] D. Yarowsky and R. Florian. Evaluating sense disambiguation performance across diverse parameter spaces. In *Journal of Natural Language Engineering*, 2003.