Word Sense Disambiguation Using Semantic Categories, Domain Information and Knowledge Sources

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research)
in
Computer Science and Engineering

by

Siva Reddy
200501107
gvsreddy@students.iiit.ac.in

Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2010

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled " Word Sense Disambiguation Using Semantic Categories, Domain Information and Knowledge Sources" by *G Venkata Sivakumar Reddy*, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Rajeev Sangal

To all my dearest ones who encouraged and supported me throughout my journey of research.

Acknowledgments

I always admired my adviser, Prof. Rajeev Sangal, whose ideals had a big influence on me which changed the way I perceived this world. I am one of those fortunate students to scribe my name in his students list. Without his support, I could not imagine myself starting a research career. His belief, “anyone can pursue research and create wonders” motivated me to contribute and made me feel to be a part of the research community. His generosity gave the freedom to enjoy all the privileges what not. I remain indebted to him all my life and just a mere thank you is not sufficient.

My existence in this world would not have been possible without my family support. I am in this position because of their love and sacrifices. I love you all.

My friend and partner of my research, Abhilash Inumella, motivated me to be strong and bold. The discussions we had, his unanswerable questions made my life tougher and helped me to bring out the best. Every word of my thesis has his contribution and this thesis can be claimed as his as well. He is the main reason to continue my research and also to apply for PhD. “Teach a man to fish; feed him for a lifetime”. This is what he does for many of us. Abhilash, you are truly inspiring.

My research life is incomplete without mentioning the people, Dr. Adam Kilgarriff and Dr. Diana McCarthy, who made me realize the best in me and also taught me research is fun. They showed me to the research world and laid a floral path to my future. Adam and Diana, you are the best and I adore you a lot. I would also like to thank Dr. Mark Stevenson who worked in collaboration with me and played a major role in getting me a PhD.

The one who insisted me on completing the thesis and move on with my future goals helped me to realize the importance of many things in my life. Spandana, you are incredible.

I extend my grateful thanks to my thesis examiner Dr. Kishore Prahallad whose comments inspired me to write a readable thesis. Initial draft and the final thesis are two extremes of north and south. I also thank my other thesis examiner Dr. Suryakanth for giving me a quick review.

I would like to thank Dr. Navjyoti Singh who believed in me, Dr. Kamalakar Karlapalem and Dr. P. J. Naryanan for institutional support, Dr. Dipti Misra Sharma and Dr. Lakshmi Bai who supported me morally, Dr. Soma Paul and Mr. Rohit Gupta who guided me in the initial stages of my research and Mr. K S Kamalakar for encouraging me in sports.

I would also like to thank Mr. Srinivas Rao for setting up a beautiful lab, Mr. Rambabu. Mr. Y. Kishore and Mr. Appaji who made my life easier, Mr. Satish, Mr. Kumara Swamy and Mr. Lakshmi Narayana who are always ready to help whenever needed and the staff of LTRC who kept our lab always neat and tidy.

I would like to acknowledge my friends and colleagues Ambati, Avinesh, Charan, Chethan, Duggi, Gani, Girish, Gopal, Harsha, Janga, John, Koneru, Kranti, Narendra, Prashant, Praveen, Pruthvi, PS, Raghudeep, Raveendra, Raz, Ravikiran, Samar, Samish, Sandeep, SCP, Siddu, Srikanth, Sriram, SRP, Viswanath, Yadav, YSP and many others who are along with me during my bad and good times. Without you I am no where buddies. I would also like thank everyone in IIIT who made my life easier.

Bibliographic Notes

Portions of this thesis are based on the following papers:

"IIITH: Domain Specific Word Sense Disambiguation". Siva Reddy, Abhilash Inumella, Diana McCarthy, Mark Stevenson. In *SemEval-2010, ACL-2010, Sweden.*

"WSD as a Distributed Constraint Optimization Problem". Siva Reddy, Abhilash Inumella. In *The Student Research Workshop, ACL-2010, Sweden*

"Hindi Sematic Category Labelling using semantic relatedness measures". Siva Reddy, Abhilash Inumella, Navjyoti Singh, Rajeev Sangal. In *The 5th International Conference of the Global WordNet Association, Mumbai, India, Jan 2010*

"All Words Unsupervised Semantic Category Labeling for Hindi". Siva Reddy, Abhilash Inumella, Rajeev Sangal, Soma Paul. In *Proceedings of Recent Advances in Natural Language Processing, Bulgaria, September, 2009*

Contents

List of Tables

List of Figures

Abstract

Words can have more than one distinct meaning and many words can be interpreted in multiple ways depending on the context in which they occur. This phenomena poses challenges to Natural Language Processing systems. State-of-the-art methods to resolve word ambiguity make use of manually annotated data. However, obtaining such data is costly for certain languages and domains. In this thesis we have developed word sense disambiguation (WSD) methods for languages and domains where no annotated data is available.

We have proposed unsupervised corpus based methods for Semantic Category Labeling, a task very similar to assigning coarse grained WSD and hence relatively easier than traditional WSD. Methods that rely on lexical knowledge base (WordNet) are also evaluated.

Furthermore, we have developed methods for domain-specific WSD and evaluated our performance on "domain specific WSD of all words" task which is a part of ACL SemEval-2010. The results reveal the importance of domain information in domain-specific WSD.

Very little work has been reported on integrating various knowledge sources for WSD. Current methods for WSD take advantage of only a few knowledge sources and do not use them collectively. We propose a novel framework which can model information from various knowledge sources into constraints and collectively use them for disambiguation. Our initial experimental results are competitive with that of state-of-the-art methods.

Chapter 1

Word Sense Disambiguation

``That Must Be Wonderful; I Have No Idea What It Means"

Words in a language may carry more than one sense. The correct sense of a word can be identified based on the context in which it occurs. In the sentence, *Any coach can instruct you to hit the ball waist high*, *coach* refers to the sense *a person who gives instruction* instead of other possibilities like *a vehicle for passengers*. The identification of the specific meaning of a word, also called as lexical disambiguation or word sense disambiguation, is rarely a problem for humans in their day to day communication, except in extreme cases. But a computer program has no basis for knowing which one is appropriate, even if it is obvious to a human. As a computational problem it is often described as "AI-complete", that is, a problem whose solution pre-supposes a solution to complete natural language understanding or common-sense reasoning.

Given a word and its possible senses, as defined in a knowledge base, the problem of Word Sense Disambiguation (WSD) can be defined as the task of assigning the most appropriate sense to the word within a given context.

WSD has been considered useful/necessary in almost every application of language technology, including machine translation (Carpuat and Wu, 2007; Resnik, 2006; Vickrey et al., 2005), information retrieval or extraction, knowledge mining or acquisition (Resnik, 2006), lexicography (Kilgarriff, 1997; Tugwell and Kilgarriff, 2001) and semantic interpretation, and is becoming increasingly important in new research areas such as bio-informatics (Weeber et al., 2001) and the Semantic Web.

Methods for WSD can be divided into knowledge based, supervised, unsupervised and semi supervised methods. Knowledge based methods (Mihalcea, 2006) mainly use dictionary knowledge like gloss overlaps (Lesk, 1986), sense relatedness measures (Patwardhan et al., 2003) or topology of senses (Agirre and Soroa, 2009). Supervised systems (Màrquez et al., 2006) learn the context in which a sense occurs from a sense tagged corpora and later use this knowledge for disambiguation. Unsupervised approaches (Yarowsky, 1992; Pedersen, 2006) exploit the notion of words having similar sense occur in similar contexts using which models from raw text are built which can disambiguate a word. Semi-supervised systems Yarowsky (1995); Tugwell and Kilgarriff (2001) use bootstrapping methods which learn knowledge from a small sense tagged corpora or rules and extend their knowledge with the already existing knowledge.

1.1 What is this thesis about?

In this thesis, we focus on the methods for coarse grained sense distinctions and domain specific sense disambiguation which do not make use of manually sense annotated corpora.

Most unsupervised corpus based methods do not use pre-existing sense inventories, but rather cluster words based on their contexts as observed in corpora and use these clusters as sense inventory. Instead we focus on using pre-existing sense inventory and propose novel corpus driven unsupervised methods for coarse grained sense distinctions.

Semantic relatedness measures have been used extensively in knowledge based methods for fine grained sense distinctions. The degree of relatedness between two senses can be measured using semantic relatedness measures. We evaluate the usefulness of six different semantic relatedness measures for coarse grained distinctions.

With the success of knowledge based methods (Agirre and Soroa, 2009) and fruitful attempts of domain specific WSD (McCarthy et al., 2004), we propose a method which takes advantage of both these approaches to perform domain specific WSD.

So far, we discussed about the usefulness of semantic relatedness measures and domain information for sense disambiguation. Apart from these, several knowledge sources are identified to be useful for WSD. Current methods for WSD take advantage of only a few knowledge sources and do not use them collectively. We propose a novel framework which

can model information from various knowledge sources into constraints and collectively use them for disambiguation.

In all our methods, we have used the knowledge bases English WordNet (Fellbaum, 1998) and Hindi WordNet (Narayan et al., 2002) as our sense inventory for the languages English and Hindi respectively.

1.2 WordNet

Senses in WordNet are described by synsets (synonyms sets). Each synset consists of a set of synonyms representing a sense defined by a gloss.

Word *coach* has the following synsets in English WordNet

```
>>>  wn.synsets('coach')
[Synset('coach.n.01'),
 Synset('coach.n.02'),
 Synset('passenger_car.n.01'),
 Synset('coach.n.04'),
 Synset('bus.n.01'),
 Synset('coach.v.01'),
 Synset('coach.v.02')]
```

Detailed description of the first synset of *coach*

```
>>> Synset('coach.n.01').synonyms
[coach, manager, handler]
>>> Synset('coach.n.01').definition
'(sports) someone in charge of training an athlete or a team'
```

In addition to providing these groups of synonyms to represent sense, WordNet connects synsets via a variety of relations. The relations provided include *antonomy*, *is-a* and *part-of*. These relations generally do not cross part of speech boundaries. For nouns, an *is-a* relation exists between two synsets when one synset *is-a-kind-of* another synset, also called *hyperynm*.

Hypernym hierarchy of first synset of *coach* is shown below.

```
>>> Synset('coach.n.01').hypernymTree
Synset('coach.n.01')
 Synset('trainer.n.01')
  Synset('leader.n.01')
   Synset('person.n.01')
    Synset('organism.n.01')
     Synset('living_thing.n.01')
      Synset('whole.n.02')
       Synset('object.n.01')
        Synset('physical_entity.n.01')
         Synset('entity.n.01')
    Synset('causal_agent.n.01')
     Synset('physical_entity.n.01')
      Synset('entity.n.01')
```

Similarly, Hindi WordNet entry for the word *billA* is shown in figures 1.1 and 1.2.

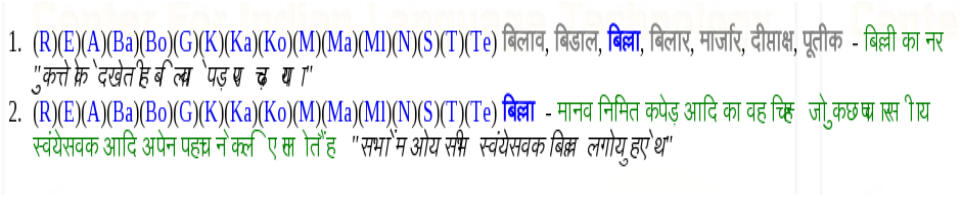

Figure 1.1: Synsets of *billA*

In Hindi WordNet, in addition to synsets it also have *ontological categories*. Ontological categories are coarse grained distinctions of word senses. These categories are organized in a hierarchical fashion based on *is-a* relation. A separate ontological hierarchy exists for each syntactic category (noun, verb, adjective adverb). Total number of categories in noun, verb, adjective and adverb hierarchy are 101, 31, 25 and 11 respectively and the maximum depth of the hierarchy is 5.

There are 28,663 synsets in Hindi WordNet. Every synset is mapped to a category in the ontological hierarchy. Figure 1.3 depicts ontological hierarchy of the word *billA*

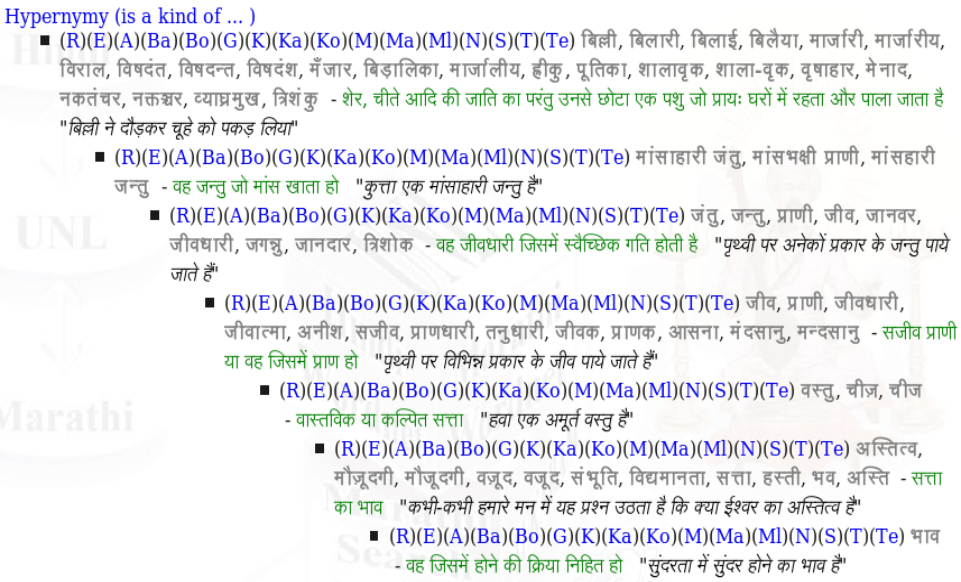

Figure 1.2: Hypernym hierarchy of first synset of $billA$

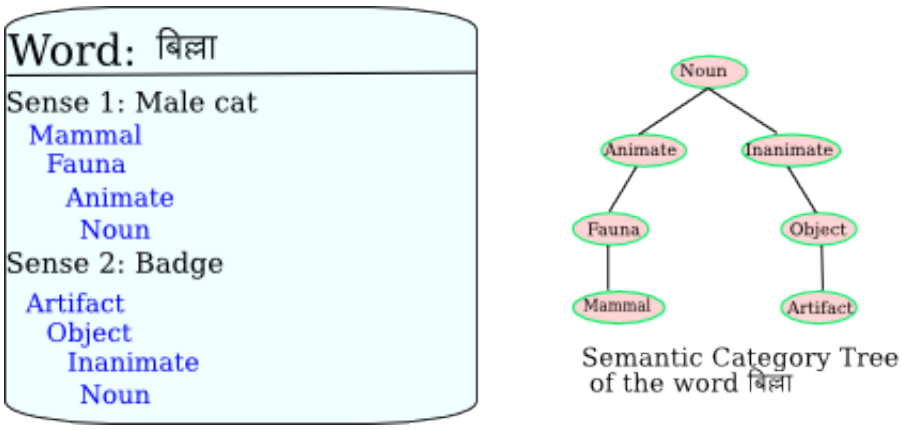

Figure 1.3: Hindi WordNet entry of the word *billA*. The word has two synsets meaning *male cat* and $badge$. Ontological category mappings of the two senses are shown on the left side of the figure. On the right, the semantic category tree(SCT) of the word is shown.

Semantic category tree (SCT) of a word is defined as the tree formed by merging the hierarchy of Ontological categories every sense of the word. Figure 1.3 shows the SCT of $billA$.

1.3 Contributions

The key contributions are four-fold. From an algorithmic point of view, we introduce novel methods for unsupervised coarse grained sense disambiguation. These methods address the problem of data sparsity and can be applied effectively to resource poor languages like Indian languages.

Secondly, we question the usefulness of semantic relatedness measures for coarse grained sense distinctions.

Furthermore, we propose a novel method for domain specific WSD which reveal the importance of domain information in sense disambiguation.

Finally, we come up with a novel framework for modelling information from various knowledge sources and collectively using this information for WSD.

1.4 Thesis overview

The remainder of the thesis is organized as follows: In the next chapter we discuss the problem of coarse grained sense disambiguation and propose unsupervised methods for it. In chapter 3, we evaluate the usefulness of semantic relatedness measures for coarse grained sense distinctions. In chapter 4 we discuss our domain specific WSD method and evaluate it. In chapter 5, we propose a novel framework for using information from various knowledge sources for WSD. Chapter 6 concludes with the main ideas and contributions of this work, along with potential directions for future research.

Chapter 2

Semantic Category Labeling

2.1 Introduction

Sense of a word can be defined as the meaning of the word, the way in which a word can be interpreted. In view of simplicity and to distinguish between a sense and a semantic category, we define semantic category of a word as its coarse grained sense. In this chapter we introduce you the task of semantic category labeling and propose two corpus driven unsupervised approaches.

Given a word, its admissible semantic categories as defined in a knowledge base and its context, the task of **semantic category labeling** (SCL) is to assign the most appropriate semantic category to the word. For the task of SCL, our language of interest is Hindi.[1] An example is shown in *Table 3.1*. We use ontological categories (section 1.2) of Hindi WordNet as our semantic category inventory . Semantic categories of the word *billA* are shown in figure 2.1.

We have chosen ontological categories of Hindi WordNet as semantic categories instead of synsets for the following reasons.

[1]Hindi is one of the official languages of India. Hindi is spoken by approximately 500 million people in the world.

1. *kuwwe/Dog* (*Mammal*) ko *xeKawe/seeing* (*NaturalEvent*) hI **billA/cat** (*Mammal*) *pedZa/tree* (*NaturalObject*) para/on caDZa/climbed (*VerbOfAction*) gayA

2. *saBA/Meeting* (*Event*) meM/in *Aye/came* (*VerbOfAction*) saBI/all *svayaMsevaka/volunteers* (*Group*) **billA/badge** (*Artifact*) lagAye/wear (*VerbOfState*) hue We

Table 2.1: Examples showing the task of semantic category labeling. *wx-notation* is used here to write Hindi.

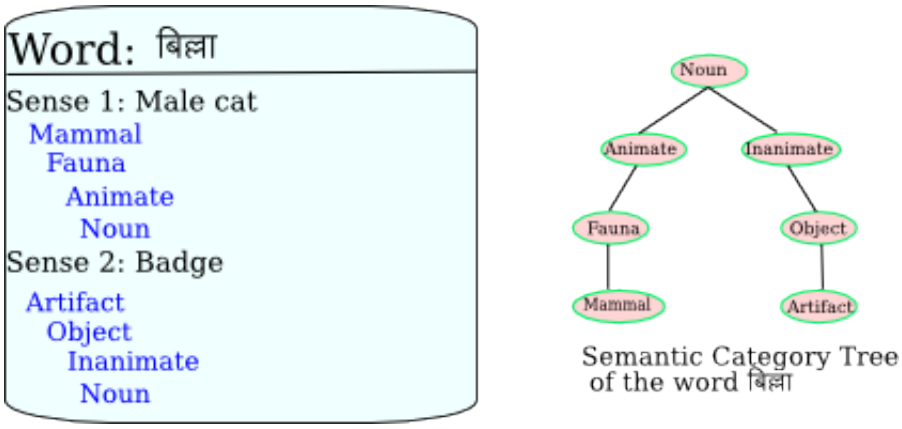

Figure 2.1: Semantic Categories of *billA* as defined in Hindi WordNet. The word has two senses meaning male cat and badge mapped to semantic categories $Mammal$ and $Artificat$. On the right, the semantic category tree (refer section 1.2) of the word is shown.

2.1.1 Ontological Categories versus Synsets?

During manual annotation of few Hindi sentences, we found that the inter annotator agreements were more when we used ontological categories compared to synsets. This is because various synsets (fine grained) are mapped to the same ontological category (coarse grained). A similar behavior was observed for English in earlier works. In the English Lexical Sample Task (Kilgarriff., 2001; Martha Palmer and Dang., 2001) of Senseval-2, the inter annotator agreement of verbs rose to 82% using the grouped senses (coarse) from 73% using WordNet 1.7 ungrouped senses. Ramakrishnan et al. (2004) states that, sense disambiguation systems should not commit to a particular sense, but rather, to a set of senses which are not necessarily orthogonal or mutually exclusive. In addition, Erk and McCarthy (2009) introduced the notion of graded word sense assignment to address sense granularity problem. With coarse grained senses, this problem does not arise often. It is also widely debated that fine grained senses are useful for humans but are not necessary for many computer applications (Ide and Wilks, 2006).

This motivates us to use ontological categories of WordNet as semantic categories.

2.1.2 Semantic Category Labeling is Useful

Recently, in the work on Hindi dependency parser by Bharati et al. (2008), the use of semantic features has been exploited. They just categories namely, human-nonhuman and animate-inanimate to boost the accuracy of dependency parser. For some labels, the increase is 5-10%. This is an encouraging result which shows the effect of using minimal semantic features on parsing accuracy.

Other tasks that can benefit from using our system are machine translation, building dictionaries from parallel corpora, named entity recognition, information extraction etc. This motivates us to present, all words unsupervised semantic category labeling.

The methodology presented here has the capability of performing both unsupervised and supervised (using sense annotated corpora) sense disambiguation. But we focus on unsupervised approach only.

In section 2.2 we present the related work. In subsection 2.3 we give the definitions. Our developed methods are presented in section 2.4. Section 2.5 covers the evaluation

aspects. To our knowledge, ours is the first attempt to work on Hindi semantic category labeling using ontological categories.

2.2 Related work

Earlier work on Hindi WSD has been done by Sinha et al. (2004) using WordNet synsets. They used adapted Lesk algorithm (Banerjee and Pedersen, 2002) where the target word's synset which has maximum overlap of its gloss, its hypernymy gloss and its hyponymy gloss with the words in the context of target word is chosen as the sense of the word. Adapted lesk cannot be effective for SCL since the definition of our semantic (ontological) categories is very general and does not have sufficient gloss to cover all its occurrences.

Patwardhan et al. (2003) WSD systems disambiguate a target word by using WordNet-based measures of semantic relatedness to find the sense of the word that is semantically most strongly related to the senses of the words in the context of the target word. Sinha and Mihalcea (2007) present Graph based unsupervised word sense disambiguation. Their work combines the word semantic similarity measures and graph centrality measures for sense disambiguation.

Most of the semantic relatedness measures between ontological categories are found to be ineffective for SCL (to be discussed in chapter 3). In this scenario, we present approaches which do not need such semantic relatedness measures.

Yarowsky (1992) unsupervised WSD system use Bayesian theoretical framework where words that are indicative to each category are identified and weighed. We use a probabilistic model with certain similarities to (Yarowsky, 1992).

2.3 Definitions

We first introduce the task formally and define some terms which are used in further discussion. The task of semantic category labeling can be formally defined as follows. Given a sequence of words $W = \{w_1, w_2, \ldots, w_n\}$ with each word w_i having semantic categories $SC_{w_i} = \{{c_{w_i}}^1, {c_{w_i}}^2, \ldots, {c_{w_i}}^{N_{w_i}}\}$, we have to assign a category to each word w_i from the set of it's semantic categories SC_{w_i}.

Definition 2.3.1. *First order collocational features* of a word *w* are the set of features describing the context of the word *w*. A feature *f* is a tuple which can be defined by one of the following templates (*sw*) or (*sw*, posOf(*sw*)) or (*sw*, posof(*sw*), posOf(*w*)) etc. where *sw* is the surrounding word of *w*, posOf(*w*) is the pos tag of *w*. Consider the following sentences.

- I ate an orange.
- Monkey is eating a banana.
- She eats an apple everyday.

If we define a feature of a word as (*sw*) within a distance of 2 words, then the first order collocational features of *eat, orange, banana* and *apple* are {*I, orange, monkey, banana, she, apple*}, {*eat*}, {*eat*}, and {*eat, everyday*} respectively. Note that words other than content words such as nouns, verbs, adjectives and adverbs are ignored while considering surrounding words.

Definition 2.3.2. *Second order collocates*: A word *x* is said to be second order collocate of *y* with respect to feature *f*, iff the feature *f* is a first order collocational feature of *x* and *y*. In the above example, {*orange, banana, apple*} are second order collocates w.r.t feature *eat* because *eat* is a first order collocational feature of all the three words *orange, banana and apple*

Definition 2.3.3. *Semantic Category Tree (SCT)*: As already said, Hindi WordNet has an ontological hierarchy and each sense of a word is mapped to some place in this hierarchy. The SCT of a word is a sub tree of this hierarchy which is shared with all the senses of this word. For example the SCT of word *billA* is shown in figure 2.1. If the pos tag of the word is known beforehand, only the sub-tree corresponding to this pos-tag is considered as SCT.

2.4 Our Approach

Our approach is inspired from the work Lin (1997). He uses *syntactic dependency* as local context to do word sense disambiguation. His work is based on the intuition that

> Two different words are likely to have similar meanings if they occur in identical local contexts.

Our assumption similar to Lin (1997) is

> Two different words are likely to have similar semantic category if they have identical first order collocational features i.e. if they are second order collocates to each other.

In this section, we present the methods Flat Semantic Category Labeler (FSCL) and Hierarchical Semantic Category Labeler (HSCL). FSCL treats semantic categories as a flat list whereas HSCL exploits the hierarchy among the categories. Both these methods take the following steps.

- **Training Phase**
  - **Step 1:** Collect second order collocates w.r.t all the features present in the training corpus.
  - **Step 2:** Build training models from second order collocation sets. Aim of this step is to calculate the likelihood of a category cat given a feature.
- **Disambiguation Phase**
  - **Step 3:** Use the above training models for Semantic Category labeling.

Detailed discussion of each step is given below.

Step 1: First order collocational features F of all the words present in the training corpus are collected using feature templates. We tried out different feature templates and the best are presented here.

1. (sw_k)
2. (sw_k , posOf(sw_k) , posOf(sw_0))

where sw_k (sw_{-k}) denote the k^{th} surrounding word to the right (left) of word of interest sw_0. k can be only in the range of ($-m$, m) where $2*m+1$ is the size of window. posOf(sw) is the part of speech tag of sw.

For every feature f_j in *F*, **S**econd **O**rder **C**ollocate sets w.r.t f_j, SOC_{f_j}, are calculated i.e. all the words which have feature f_j as first order collocational feature are collected.

In the following sections, we discuss two different methods that differ in the way *steps 2,3* are performed.

2.4.1 Flat Semantic Category labeler (FSCL)

Based on our assumption that second order collocates w.r.t a feature f_j, SOC_{f_j}, are likely to have same semantic category, we calculate the expectation of the occurrence of each semantic category with feature f_j. Only the leaf semantic categories of the all the words in the second order collocate set SOC_{f_j} are considered. Hierarchical information of the semantic categories is not used.

Step 2: Aim of this step is to calculate the expectation of occurrence of category cat with feature f_j. To calculate this, we use the following equations.

$$\begin{aligned} Pr(cat|f_j) &= \frac{Count(cat, SOC_{f_j})}{\sum_{cat} Count(cat, SOC_{f_j})} \\ AE(cat|f_j) &= \frac{Pr(cat|f_j)}{Pr(f_j)} \end{aligned} \tag{2.1}$$

where $Pr(cat|f_j)$ denotes the probability of the occurrence of category cat with feature f_j. $Count(cat, SOC_{f_j})$ denotes the number of words in SOC_{f_j} which have category cat as their leaf semantic category. $AE(cat/f_j)$ is the above expectation measure which gives the expectation of occurrence of cat with feature f_j. Some of the most frequent features consisting of function words, occur with almost all the categories. This measure penalizes such words and rewards the salient features of a category. A similar AE measure can be found in Kavalec et al. (2004). In his work, the measure *above expectation* (AE) is employed for non taxonomic relation extraction.

Step 3: This is the disambiguation phase where an utterance of a word w_i with leaf semantic categories SC_{w_i}= $\{c_1, c_2, \ldots\}$ is assigned a category according to the following

equation.

$$\underset{c_k \in SC_{w_i}}{\operatorname{argmax}} \sum_{j=1}^{F} AE(c_k|f_j)$$

where $f_1, f_2, \ldots, f_F$ are the first order collocational features of w_i. $AE(c_k|f_j)$ is the expectation of the occurrence of c_k with feature f_j which is calculated in the previous step (training phase). The expected occurrence of each admissible leaf category of w_i is calculated w.r.t all its first order collocational features and the category with highest score is chosen. We used summation over all features because AE is an expectation measure and not a probability measure.

2.4.2 Hierarchical Semantic Category labeler (HSCL)

This method uses the hierarchical information of the semantic category tree (FSCL uses only leaf categories). In the training phase, given a feature, the expectation of each category at each level of the semantic category tree are calculated. The disambiguation algorithm runs in a top down fashion and takes a decision at each level based on the available expectation scores at that level. The details are as follows.

Step 2: Given a feature f_j, the aim of this step is to calculate the expectation of each category at each level of the semantic category tree (SCT). Let c_h^i denote i^{th} category at the level h of SCT. This phase is summarized below.

- Aggregate the semantic category trees of all of the words in the set SOC_{f_j}: The semantic category trees of the second order collocates w.r.t feature f_j are obtained. They are aggregated in this step to form the aggregate tree AGT_{f_j}. To perform the aggregation we take the union of semantic category trees of all the words in SOC_{f_i}. Union of two trees is a simple operation by doing which, the nodes common to both the trees get their scores summed up and the for others it remains the same. Initially each tree node carries a score of .

$$node.score = \frac{1}{|node.siblings| + 1}$$

Aggregation of trees:

$$AGT_{f_j} = \{ \bigcup_{w \in SOC_{f_j}} SCT(w) \}$$

- Normalize the scores of each node in the tree AGT_{f_j} to calculate $Pr(c_h^i|f_j)$: The nodes in of the tree AGT_{f_j} carry the summed up scores as a result of aggregation operation performed in previous step. These scores are normalized according the following equation.

$$\begin{aligned} Pr(c_h^i|f_j) &\simeq \frac{n_h^i.score}{|SOC_{f_j}|} \\ AE(c_h^i|f_j) &= \frac{Pr(c_h^i|f_j)}{Pr(f_j)} \end{aligned} \tag{2.2}$$

  where n_h^i is the node in AGT_{f_j} corresponding to the category c_h^i. $AE(c_h^i|f_j)$ is the above expectation measure which gives the expectation of the occurrence of c_h^i when the feature f_j. Note: $Pr(c_h^i|f_j)$ is not the exact probability. This measure gives more preference to the words in SOC_{f_j} which are less ambiguous.

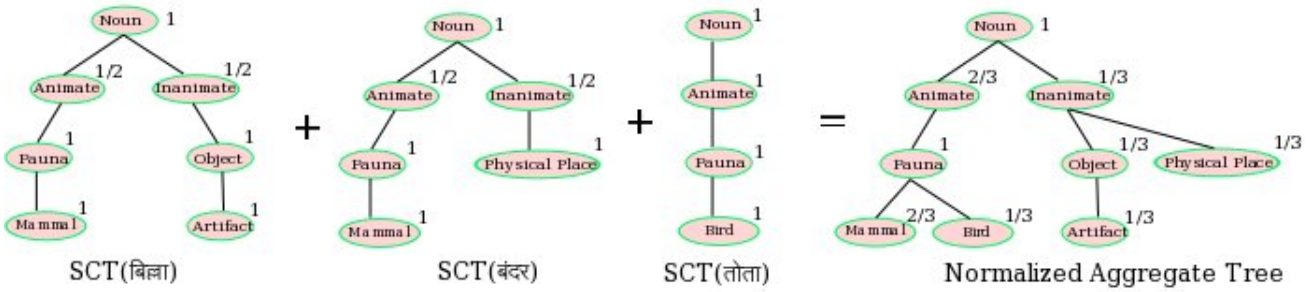

Figure 2.2: Aggregation and Normalization of Semantic Category trees

The example shown in the figure 2.2 clarifies the aggregation and normalization steps used in this algorithm. The feature used in this example is (*caDZa/climb*). The set $SOC_{\text{(caDZa/climb)}}$ consists of words *billA/Cat, baMxara/Monkey, wowA/Parrot*. The semantic category trees of these words are shown on the left with their initial scores. The right most tree is formed after the normalization of the aggregated tree $AGT_{\text{(caDZa/climb)}}$.

In this paragraph, we discuss an alternative scoring mechanism. The same example in figure 2.2 is used to explain this mechanism. The probabilities in this are calculated as

follows. Take initial $node.score$ to be 1 for each tree. Aggregate all of them to form an aggregate tree. In the example figure, scores on nodes $Noun$, $Animate$, $Inanimate$ of the aggregate tree will be 3, 3, 2 respectively. Normalization is performed using the following equation

$$Pr(c_h^i|f_j) \ = \ \frac{n_h^i.score}{\sum_k n_h^k.score}$$

Scores on nodes $Noun$, $Animate$, $Inanimate$ of the aggregate tree after the normalization are $3/3, 3/5, 2/5$ respectively. The ratio of the probability of $Animate$ and $Inanimate$ is $(3/5)/(2/5)$ = $3/2$. Using the former scoring mechanism it is $(2/3)/(1/3)$ = 2. The former mechanism accumulates a higher confidence for the category $Animate$ compared to $Inanimate$ because it gives more preference to words with one sense (here wowA/parrot) and the latter model gives equal preference to all the words. To put it in other words, our scoring mechanism gives preference to semantic category of the words with single sense assuming that this semantic category is more likely to occur with the given feature.

Step 3: To disambiguate an occurrence a word w_i with it's collocational features f_j a top down walk is performed on the semantic category tree of w_i. The set of categories at level h (denoted by $SCT^h(w_i)$) are disambiguated first before moving to disambiguate at level $h+1$. Once a category is decided at level h, then the algorithm considers only the children of this category in level $h+1$. This results in reducing the semantic category search space of disambiguation algorithm. For more details, refer to the algorithm below.

Algorithm 1 HSCL Disambiguation phase

1: Input: w_i and it's collocational features f_j
2: Output: A semantic category path.
3: cur=TOP
4: **for** each level $h \in \{0, 1, \ldots\}$ **do**
5: $pList = \{c|c \in SCT^h(w_i) \& parent(c) == cur\}$ //pruning the list of categories at level h
6: $cur = \underset{cat \in pList}{\text{argmax}} \sum_{j=1}^{F} AE(cat|f_j)$
7: append cur to output
8: **end for**

Advantages of HSCL:

- HSCL disambiguate level by level. Number of categories to be disambiguated in the

top level are less compared to the number of leaves of the semantic category tree. This reduces the search space while disambiguation and hence it becomes simpler.

- No need of semantic similarity/relatedness measures.
- The nodes at top levels are shared by large number of words. This makes the learning effective for these nodes and hence the method takes better decisions at top levels.
- This can handle unseen category instances because the disambiguation proceeds in top down manner.
- This method can stop at a level which has high confidence score.

2.5 Evaluation

We trained our methods on a 1.2 million word corpus. We used a separate corpus for evaluating the proposed algorithms. The testing data comprises of 7200 manual annotated sentences which cover 133 semantic categories.

It is desirable to have high precision and low recall systems in certain scenarios. To achieve this, a word is committed to a category only if the confidence score is greater than the set threshold value. The threshold value is chosen to be the k times the average of the set S consisting of all category scores over all the features.

$$\theta = k * averageoftheSet(S)$$

where $\forall cat \forall j Pr(cat|f_j) \in S$. Set S is collected during Training phase. As k is increased, precision increases (with decrease in recall)

2.5.1 FSCL accuracies

The **baseline** system assigns the semantic category of first sense of the word. The evaluation results of FSCL for **nouns** is shown in table 2.2.

Model	**P**	**R**
Baseline	85.6	85.6
FSCL trained on raw text	75.6	75.6
FSCL with k=2 trained on raw text	84.7	53.9
FSCL with k=3 trained on raw text	87.8	50.0
FSCL with k=2 trained on pos tagged text	83.2	63.4

Table 2.2: Accuracies of FSCL and Baseline for **nouns** (P: precision and R:recall)

As discussed in *Section 4*, feature (sw_k) and (sw_j, $pos(sw_j)$, $pos(sw_0)$) are used as features for training on raw and pos-tagged text respectively. Window size of 20 is used in all the models.

As k value is increased, FSCL method performs better than the baseline. We believe that the reason for low recall is because of the size of training corpus. For English, huge corpora above 100 million words are available. But for Hindi, such huge corpora does not exist. Once if our models are built using such huge corpora, recall can also be increased since the number of salient words for each category increases.

We see that the precision of the model trained on pos-tagged text is less compared to others because of the low accuracy of the Hindi pos-tagger which is about 78%.[2] Training corpus is pos tagged using Avinesh.PVS and Gali (2007).

2.5.2 Level wise accuracies of HSCL

	Baseline		HSCL ($k = 5$)	
Level	P	R	P	R
1	96.9	96.9	99.4	94.0
2	91.5	91.5	96.4	63.8
3	89.8	89.8	95.4	52.0
4	87.7	87.7	94.4	46.4
5	76.8	76.8	83.1	64.4

Table 2.3: Level wise accuracies of HSCL for **nouns**

For each level, the **baseline** system assigns the semantic category of the first sense of

[2]Accuracy of the pos tagger has improved since then.

the word corresponding to that level.

The results obtained using HSCL method with $k = 5$ are shown in the table 2.3. Window size of 20 is taken. Raw text is used for training. We see that HSCL outperforms the baseline (first sense) in terms of precision. The recall values of HSCL are low compared to baseline.

Comparing HSCL with FSCL, precision values of HSCL are very high and the recall values of HSCL are comparable with FSCL. This shows us that high precision values can be achieved with HSCL compared to FSCL for the same recall values.

2.6 Summary

We have introduced the problem of SCL and also presented two unsupervised methods for performing this task. These methods do not rely on semantic similarity measures. We also presented the advantages of top-down approach called HSCL. To label an utterance of size n, an efficient implementation of our disambiguation procedure takes a time $O(n * s)$, where s is the maximum number of senses of a word in this utterance. Besides presenting the evaluations of our algorithms, we also presented a simple parameter tuning procedure to obtain a precision recall trade off.

Future work remains to integrate our system with Hindi dependency parser and study the effect of semantic features on parsing accuracies. It will be interesting to apply the methods discussed in to English language using the synset hierarchy of English WordNet.

Chapter 3

Evaluation of Semantic Relatedness Measures

3.1 Introduction

The task of semantic category labeling (SCL) has been introduced in chapter 2. Given a word, its admissible semantic categories as defined in a knowledge base and its context, the task of SCL is to assign the most appropriate semantic category to the word. An example is shown in Table 3.1.

In chapter 2, we have introduced unsupervised methods for SCL. In this chapter we evaluate the usefulness of semantic relatedness measures for SCL. Semantic relatedness

1. *kuwwe*/*Dog* (*Mammal*) ko *xeKawe*/*seeing* (*NaturalEvent*) hI **billA**/**cat** (*Mammal*) *pedZa*/*tree* (*NaturalObject*) para/on caDZa/climbed (*VerbOfAction*) gayA
2. *saBA*/*Meeting* (*Event*) meM/in *Aye*/*came* (*VerbOfAction*) saBI/all *svayaMsevaka*/*volunteers* (*Group*) **billA**/**badge** (*Artifact*) lagAye/wear (*VerbOfState*) hue We

Table 3.1: Examples showing the task of semantic category labeling. *wx-notation* is used here to write Hindi.

is a measure of how related two or more concepts are. There are a number of semantic relatedness measures proposed on WordNet which make use of sense definitions and sense hierarchy.

3.2 Related Work

Inspired by the original Lesk algorithm (Lesk, 1986), a number of WordNet based disambiguation algorithms were proposed. Lesk algorithm disambiguate a target word by assigning the sense whose gloss (definition) maximally overlaps with the neighbouring words gloss. Banerjee and Pedersen (2002) used hierarchical relationships in WordNet to include the glosses of words that are related to the target word and its neighbours. Patwardhan et al. (2003) takes the view that gloss overlaps are just another measure of semantic relatedness. They evaluated a number of semantic relatedness measures for English word sense disambiguation.

Our work builds on the earlier work of (Patwardhan et al., 2003). We evaluated a number of semantic relatedness measures in the light of Hindi Semantic Category Labeling.

3.3 Experimental Setting

3.3.1 Labeling Algorithm

We used a variation of simplified Lesk algorithm to label the semantic category for a given target word in a given context. Unlike simplified Lesk algorithm which uses gloss overlap of target word's category and sentential context as a relatedness measure, our algorithm generalizes by using other semantic relatedness measures. It chooses the semantic category of the target word which is maximally related with its context. We used the immediate neighbours of the target word W_T, word to the left W_L and word to the right W_R, as context of the target word. The semantic category 'C' which maximizes the following equation is chosen to be the label of the target word.

$$\max_{C \in cat(W_T)} (leftRelatedness(C) +$$

$$rightRelatedness(C)\ )$$

where

cat(w) are the categories of the word 'w'

$$leftRelatedness(C) = \max_{L\ \in\ cat(W_L)} Rel(C, L)$$

$$rightRelatedness(C) = \max_{R\ \in\ cat(W_R)} Rel(C, R)$$

and *Rel() gives the semantic relatedness value between two categories measured using semantic relatedness measure*

The next section describes the semantic relatedness measures used by us.

3.3.2 Semantic Relatedness Measures

Semantic Relatedness Measure gives a metric to measure the relatedness of two concepts. A concept can either refer to a semantic category or a synset. We conducted our evaluation using the following Semantic relatedness metrics: Lesk (lesk), adapted Lesk (adpLesk), Leacock & Chodorow (lch), Wu & Palmer (wup), Lin (lin), and Jiang & Conrath (jcn). We provide below a short description for each of these six metrics.

We view gloss overlaps as just another measure of semantic relatedness. Simplified and adapted Lesk relatedness measures are based on this assumption.

The Lesk Measure

Lesk relatedness between two concepts is the number of gloss overlaps of the two concepts. Hindi WordNet Ontological categories does not contain adequate gloss (and examples). To provide more gloss for an ontological category, we used the gloss of the synsets which correspond to this ontological category.

$$Rel_{lesk} = Overlap\ (\ gloss_{concept1},\ gloss_{concept2}\ )$$

The Adapted Lesk Measure

Adapted Lesk relatedness between two concepts is defined as

$$Rel_{adpLesk} =$$

$$Overlap\ (\ extendedGloss_{concept1},$$

$$extendedGloss_{concept2}\ )$$

where extendedGloss of a concept is the total gloss of all the concepts in the hierarchy of the given concept (including itself).

The Leacock-Chodorow Measure

The Leacock and Chodorow (1998) relatedness between two concepts is determined as:

$$Rel_{lch} = -log\frac{length}{2D}$$

where length is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of the hierarchy.

The Wu-Palmer Measure

The Wu and Palmer (1994) relatedness metric measures the depth of two given concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a relatedness score:

$$Rel_{wup} = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)}$$

The Lin Measure

The Lin (1998b) relatedness between two concepts is defined as

$$Rel_{lin} = \frac{2 * IC(LCS)}{IC(concept_1) + IC(concept_2)}$$

where IC is defined as:

$$IC(c) = -logP(c)$$

and P (c) is the probability of encountering an instance of concept c in a large corpus. As we don't have a large sense tagged corpora, we calculated this probability by making the assumption done in (Patwardhan et al., 2003): each category of a word is equally likely. We used a Hindi web corpora of size 324 MB to collect these statistics.

The Jiang-Conrath Measure

The Jiang and Conrath (1997) Measure used by us is similar to the one used in (Patwardhan et al., 2003)

$$Rel_{jcn} = \frac{1}{IC(concept_1) + IC(concept_2) - 2 * IC(LCS)}$$

3.4 Evaluation

3.4.1 Data

In our experiments, we labeled only nouns. We evaluated our experiments on manually annotated sense (semantic category and synset) tagged data developed by Indian language machine translation consortium(ILMT). It comprises of articles from news and tourism domain. In all, there are 7200 manual annotated sentences covering 133 semantic categories. The average semantic category ambiguity of a word is 2.18 i.e. on an average each word can have 2.18 semantic categories whereas synset ambiguity of a word is 2.57.

3.4.2 Results

The results of the experiment are shown in tables 3.2 and 3.3. In table 3.2, the accuracies for the task of semantic category labeling are shown and in table 3.3, the results for word sense disambiguation using synsets are shown. As we can see from the results the semantic categories are coarse grained and hence it turns out to be an easier task compared to synset

Model	**Precision**	**Recall**	**F-Measure**
Baseline	84.76	84.76	84.76
lch	74.87	61.30	67.40
wup	75.33	61.68	67.82
lin	74.11	60.17	66.41
jcn	71.93	51.43	59.97
lesk	74.75	72.73	73.72
adpLesk	76.05	74.09	**75.05**

Table 3.2: Evaluation of Semantic Category Labeling of Nouns

Model	**Precision**	**Recall**	**F-Measure**
Baseline	78.23	78.23	78.23
lch	67.14	54.98	60.45
wup	67.45	55.23	60.73
lin	65.05	52.81	58.29
jcn	62.52	44.70	52.12
lesk	65.27	63.51	**64.37**
adpLesk	63.36	61.73	62.53

Table 3.3: Evaluation of Synset Assignment of Nouns

assignment. This is expected because in a number of cases multiple synsets correspond to same semantic category in Hindi WordNet.

Also, the accuracies of baseline system, which assigns the first sense is considerably higher than others. WordNet senses are listed in the order of its frequency from which the sense inventory is created. Our testing corpus might have fallen along these lines of WordNet creation. This might be the reason of having higher accuracies for baseline. On a different corpus(domain), the first sense in WordNet might not be the frequent sense in the domain of interest. This effects the accuracy of the baseline system. This is not the case with the algorithms based on relatedness measures.

3.4.3 Observations

It is interesting to observe that adapted Lesk performs well on semantic category labeling than on synset labeling whereas Lesk performs well on synset labeling. This gives an insight that gloss information of Hindi WordNet ontological category is not sufficient for semantic category labeling and has to depend on the ontological hierarchy. In the case of synsets, the gloss information is adequate and the addition of hierarchical information may create noise.

Results show that Lesk and adapted Lesk are performing well (F-measure) compared to other semantic relatedness measures. This might be due to the reason that Lesk and adapted Lesk can relate two words across the syntactic categories (part-of-speech (pos) tags) which is not the case with other relatedness measures used in this work. It can be inferred from the results that SCL and WSD benefits from relatedness measures which can relate words across pos categories. Apart from Lesk and adapted Lesk measures, rest of the measures are more of similarity measures than relatedness measures.

3.5 Summary

In this chapter , we have evaluated the usefulness of a number of semantic relatedness measures for semantic category labeling and synset disambiguation. SCL or WSD benefits from relatedness measures like Lesk and adapted Lesk rather than similarity measures. For

the task of semantic category labeling, the measure **adapted Lesk** performs better than all other measures.

Chapter 4

Domain Specific WSD

4.1 Introduction

In chapter 3 we have seen the effectiveness of semantic relatedness for disambiguation. In this chapter we discuss the importance of domain information in disambiguation.

The senses in WordNet are ordered according to their frequency in a manually tagged corpus, SemCor (Miller et al., 1993). Senses that do not occur in SemCor are ordered arbitrarily after those senses of the word that have occurred. It is known from the results of SENSEVAL2 (Cotton et al., 2001) and SENSEVAL3 (Mihalcea and Edmonds, 2004) that first sense heuristic outperforms many WSD systems (see McCarthy et al. (2007)). The first sense baseline's strong performance is due to the skewed frequency distribution of word senses. WordNet sense distributions based on SemCor are clearly useful, however in a given domain these distributions may not hold true. For example, the first sense for "bank" in WordNet refers to "sloping land beside a body of river" and the second to "financial institution", but in the domain of "finance" the "financial institution" sense would be expected to be more likely than the "sloping land beside a body of river" sense. Unfortunately, it is not feasible to produce large manually sense-annotated corpora for every domain of interest. McCarthy et al. (2004) propose a method to predict sense distributions from raw corpora and use this as a first sense heuristic for tagging text with the predominant sense. Rather than assigning predominant sense in every case, our approach aims to use these sense distributions collected from domain specific corpora as a knowledge source

and combine this with information from the context.

Our approach focuses on the strong influence of domain for WSD (Buitelaar et al., 2006) and the benefits of focusing on words salient to the domain (Koeling et al., 2005). Words are assigned a ranking score based on its keyness (salience) in the given domain. We use these word scores as another knowledge source.

Graph based methods have been shown to produce state-of-the-art performance for unsupervised word sense disambiguation (Agirre and Soroa, 2009; Sinha and Mihalcea, 2007). These approaches use well-known graph-based techniques to find and exploit the structural properties of the graph underlying a particular lexical knowledge base (LKB), such as WordNet. These graph-based algorithms are appealing because they take into account information drawn from the entire graph as well as from the given context, making them superior to other approaches that rely only on local information individually derived for each word.

Our approach uses the Personalized PageRank algorithm (Agirre and Soroa, 2009) over a graph representing WordNet to disambiguate ambiguous words by taking their context into consideration. We also combine domain-specific information from the knowledge sources, like sense distribution scores and keyword ranking scores, into the graph thus personalizing the graph for the given domain.

In section 4.2, we describe domain sense ranking. Domain keyword ranking is described in Section 4.3. Graph construction and personalized page rank are described in Section 4.4. Evaluation results over the SemEval data are provided in Section 4.5.

4.2 Domain Sense Ranking

McCarthy et al. (2004) propose a method for finding predominant senses from raw text. The method uses a thesaurus acquired from automatically parsed text based on the method described by Lin (1998a). This provides the top k nearest neighbours for each target word w, along with the distributional similarity score between the target word and each neighbour. The senses of a word w are each assigned a score by summing over the distributional similarity scores of its neighbours. These are weighted by a semantic similarity score (using WordNet Similarity score (Pedersen et al., 2004) between the sense of w and the sense

of the neighbour that maximizes the semantic similarity score.

More formally, let $N_w = \{n_1, n_2, \ldots n_k\}$ be the ordered set of the top k scoring neighbours of w from the thesaurus with associated distributional similarity scores $\{dss(w, n_1), dss(w, n_2), \ldots dss(w, n_k)\}$. Let $senses(w)$ be the set of senses of w. For each sense of w ($ws_i \in senses(w)$) a ranking score is obtained by summing over the $dss(w, n_j)$ of each neighbour ($n_j \in N_w$) multiplied by a weight. This weight is the WordNet similarity score ($wnss$) between the target sense (ws_i) and the sense of n_j ($ns_x \in senses(n_j)$) that maximizes this score, divided by the sum of all such WordNet similarity scores for $senses(w)$ and n_j. Each sense $ws_i \in senses(w)$ is given a sense ranking score $srs(ws_i)$ using

$$srs(ws_i) = \sum_{n_j \epsilon N_w} dss(w, n_j) \times \frac{wnss(ws_i, n_j)}{\sum_{ws_i \epsilon senses(w)} wnss(ws_i, n_j)}$$

where $wnss(ws_i, n_j) =$

$$max_{ns_x \in senses(n_j)}(wnss(ws_i, ns_x))$$

Since this approach requires only raw text, sense rankings for a particular domain can be generated by simply training the algorithm using a corpus representing that domain. We used the background documents provided to the participants in this task as a domain specific corpus. In general, a domain specific corpus can be obtained using domain-specific keywords (Kilgarriff et al., 2010). A thesaurus is acquired from automatically parsed background documents using the Stanford Parser (Klein and Manning, 2003). We used $k = 5$ to built the thesaurus. As we increased k we found the number of non-domain specific words occurring in the thesaurus increased and negatively affected the sense distributions. To counter this, one of our systems IIITH2 used a slightly modified ranking score by multiplying the effect of each neighbour with its domain keyword ranking score. The modified sense ranking $msrs(ws_j)$ score of sense ws_i is

$$msrs(ws_i) =$$

$$\sum_{n_j \epsilon N_w} dss(w, n_j) \times \frac{wnss(ws_i, n_j)}{\sum_{ws_i \epsilon senses(w)} wnss(ws_i, n_j)} \times krs(n_j)$$

where $krs(n_j)$ is the keyword ranking score of the neighbour n_j in the domain specific corpus. In the next section we describe the way in which we compute $krs(n_j)$.

WordNet::Similarity::lesk (Pedersen et al., 2004) was used to compute word similarity wnss. IIITH1 and IIITH2 systems differ in the way senses are ranked. IIITH1 uses $srs(ws_j)$ whereas IIITH2 system uses $msrs(ws_j)$ for computing sense ranking scores in the given domain.

4.3 Domain Keyword Ranking

We extracted keywords in the domain by comparing the frequency lists of domain corpora (background documents) and a very large general corpus, ukWaC (Ferraresi et al., 2008), using the method described by Rayson and Garside (2000). For each word in the frequency list of the domain corpora, words(domain), we calculated the log-likelihood (LL) statistic as described in Rayson and Garside (2000). We then normalized LL to compute keyword ranking score $krs(w)$ of word $w\ words(domain)$ using

$$krs(w) = \frac{LL(w)}{\sum_{w_i \in words(domain)} LL(w_i)}$$

The above score represents the keyness of the word in the given domain. Top ten keywords (in descending order of krs) in the corpora provided for this task are *species, biodiversity, life, habitat, natura*[1]*, EU, forest, conservation, years, amp*[2].

[1]In background documents this word occurs in reports describing Natura 2000 networking programme.

[2]This new word *"amp"* is created by our programs while extracting body text from background documents. The HTML code *"&"* which represents the symbol*"&"* is converted into this word.

4.4 Personalized PageRank

Our approach uses the Personalized PageRank algorithm (Agirre and Soroa, 2009) with WordNet as the lexical knowledge base (LKB) to perform WSD. WordNet is converted to a graph by representing each synset as a node (synset node) and the relationships in WordNet (hypernymy, hyponymy etc.) as edges between synset nodes. The graph is initialized by adding a node (word node) for each context word of the target word (including itself) thus creating a context dependent graph (personalized graph). The popular PageRank (Page et al., 1999) algorithm is employed to analyze this personalized graph (thus the algorithm is referred as personalized PageRank algorithm) and the sense for each disambiguous word is chosen by choosing the synset node which gets the highest weight after a certain number of iterations of PageRank algorithm.

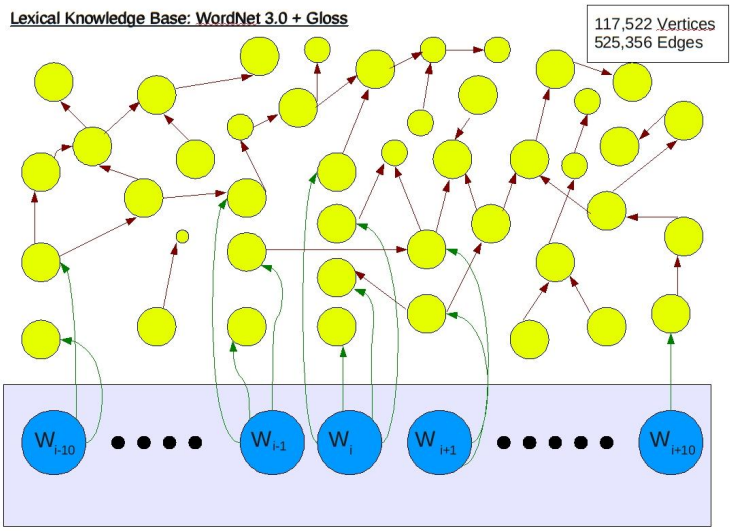

Figure 4.1: Personalized Graph: Yellow and Blue nodes represent synsets and context words respectively. Green edges connecting the word and its synsets are assigned weights equal to Sense Ranking Scores (srs). Blue nodes are initialized with Keyword Ranking Scores (krs).

We capture domain information in the personalized graph by using sense ranking scores and keyword ranking scores of the domain to assign initial weights to the word nodes and their edges (word-synset edge). This way we personalize the graph for the given domain. Figure 4.1 depicts a personalized graph.

4.4.1 Graph Initialization Methods

	Precision	**Recall**
Unsupervised Graph Initialization		
PPR	37.3	36.8
KRS + PPR	38.1	37.6
SRS + PPR	48.4	47.8
KRS + SRS + PPR	48.0	47.4
Semi-supervised Graph Initialization		
CS + PPR	50.2	49.6
CS + KRS + PPR	50.1	49.5
* CS + SRS + PPR	53.4	52.8
CS + KRS + SRS + PPR	**53.6**	**52.9**
Others		
1^{st}sense	50.5	50.5
PSH	49.8	43.2

Table 4.1: Evaluation results on English test data of SemEval-2010 Task-17. * represents the system which we submitted to SemEval and is ranked 3rd in public evaluation.

We experimented with different ways of initializing the graph, described below, which are designed to capture domain specific information.

Personalized Page rank (PPR): In this method, the graph is initialized by allocating equal probability mass to all the word nodes in the context including the target word itself, thus making the graph context sensitive. This does not include domain specific information.

Keyword Ranking scores with PPR (KRS + PPR): This is same as PPR except that context words are initialized with krs.

Sense Ranking scores with PPR (SRS + PPR): Edges connecting words and their synsets are assigned weights equal to srs. The initialization of word nodes is same as in PPR.

KRS + SRS + PPR: Word nodes are initialized with krs and edges are assigned weights equal to srs.

In addition to the above methods of unsupervised graph initialization, we also initialized the graph in a *semi-supervised* manner. WordNet (version 1.7 and above) have a field *tag_cnt* for each synset (in the file *index.sense*) which represents the number of times the

synset is tagged in various semantic concordance texts. We used this information, *concordance score* (cs) of each synset, with the above methods of graph initialization as described below.

Concordance scores with PPR (CS + PPR): The graph initialization is similar to PPR initialization additionally with concordance score of synsets on the edges joining words and their synsets.

CS + KRS + PPR: The initialization graph of KRS + PPR is further initialized by assigning concordance scores to the edges connecting words and their synsets.

CS + SRS + PPR: Edges connecting words and their synsets are assigned weights equal to sum of the concordance scores and sense ranking scores i.e. $cs + srs$. The initialization of word nodes is same as in PPR.

CS + KRS + SRS + PPR: Word nodes are initialized with krs and edges are assigned weights equal to $cs + srs$.

PageRank was applied to all the above graphs to disambiguate a target word.

4.4.2 Experimental details of PageRank

Tool: We used UKB tool[3] (Agirre and Soroa, 2009) which provides an implementation of personalized PageRank. We modified it to incorporate our methods of graph initialization. The LKB used in our experiments is WordNet3.0 + Gloss which is provided in the tool. More details of the tools used can be found in the Appendix-1.

Normalizations: Sense ranking scores (srs) and keyword ranking scores (krs) have diverse ranges. We found srs generally in the range between 0 to 1 and krs in the range 0 to 0.02. Since these scores are used to assign initial weights in the graph, these ranges are scaled to fall in a common range of [0, 100]. Using any other scaling method should not effect the performance much since PageRank (and UKB tool) has its own internal mechanisms to normalize the weights.

[3] `http://ixa2.si.ehu.es/ukb/`

4.5 Evaluation Results

Test data released for this task is disambiguated using IIITH1 and IIITH2 systems. As described in Section 2, IIITH1 and IIITH2 systems differ in the way the sense ranking scores are computed. Here we project only the results of IIITH1 since IIITH1 performed slightly better than IIITH2 in all the above settings. Results of 1^{st}*sense* system provided by the organizers which assigns first sense computed from the annotations in hand-labeled corpora is also presented. Additionally, we also present the results of Predominant Sense Heuristic (PSH) which assigns every word w with the sense ws_j ($ws_j \in senses(w)$) which has the highest value of $srs(ws_j)$ computed in Section 2 similar to (McCarthy et al., 2004).

Table 5.1 presents the evaluation results. We used TreeTagger [4] to Part of Speech tag the test data. POS information was used to discard irrelevant senses. Due to POS tagging errors, our precision values were not equal to recall values. In the competition, we submitted IIITH1 and IIITH2 systems with CS + SRS + PPR graph initialization. IIITH1 and IIIH2 gave performances of 53.4 % and 52.2 % precision respectively. In our later experiments, we found CS + KRS + SRS + PPR has given the best performance of 53.6 % precision.

From the results, it can be seen when srs information is incorporated in the graph, precision improved by 11.1% compared to PPR in unsupervised graph initialization and by 3.19% compared to CS + PPR in semi-supervised graph initialization. Also little improvements are seen when krs information is added. This shows that domain specific information like sense ranking scores and keyword ranking scores play a major role in domain specific WSD.

The difference between the results in unsupervised and semi-supervised graph initializations may be attributed to the additional information the semi-supervised graph is having i.e. the sense distribution knowledge of non-domain specific words (common words).

4.6 Summary

We have proposed a method for domain specific WSD. The results show that domain information play a crucial role and greater improvements can be achieved incorporating domain

[4] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

information into wsd methods. Sense ranking scores are found to be more effective than keyword ranking scores.

Chapter 5

Framework for knowledge source interactions

5.1 Introduction

In chapter 3 and 4 we have explored the usefulness of semantic relatedness measures and domain information for sense disambiguation. In this chapter we propose a framework which aim to use information from many such knowledge sources.

WSD is one of the oldest problems in computational linguistics which dates back to early 1950's. A range of knowledge sources have been found to be useful for WSD. (Agirre and Stevenson, 2006; Agirre and Martínez, 2001; McRoy, 1992; Hirst, 1987) highlight the importance of various knowledge sources like part of speech, morphology, collocations, lexical knowledge base (sense taxonomy, gloss), sub-categorization, semantic word associations, selectional preferences, semantic roles, domain, topical word associations, frequency of senses, collocations, domain knowledge. etc. Methods for WSD exploit information from one or more of these knowledge sources.

Supervised approaches like (Yarowsky and Florian, 2002; Lee and Ng, 2002; Martínez et al., 2002; Stevenson and Wilks, 2001) used collective information from various knowledge sources to perform disambiguation. Information from various knowledge sources is encoded in the form of a feature vector and models were built by training on sense-tagged corpora. These approaches pose WSD as a classification problem. They crucially rely on

hand-tagged sense corpora which is hard to obtain. Systems that do not need hand-tagging have also been proposed. Agirre and Martínez (2001) evaluated the contribution of each knowledge source separately. However, this does not combine information from more than one knowledge source.

In any case, little effort has been made in formalizing the way in which information from various knowledge sources can be collectively used within a single framework: a framework that allows interaction of evidence from various knowledge sources to arrive at a global optimal solution.

Here we present a way for modelling information from various knowledge sources in a multi agent setting called distributed constraint optimization problem (DCOP). In DCOP, agents have constraints on their values and each constraint has a utility associated with it. The agents communicate with each other and choose values such that a global optimum solution (maximum utility) is attained. We aim to solve WSD by modelling it as a DCOP.

To the best of our knowledge, ours is the first attempt to model WSD as a DCOP. In DCOP framework, information from various knowledge sources can be used combinedly to perform WSD.

In section 5.2, we give a brief introduction of DCOP. Section 5.3 describes modelling WSD as a DCOP. Utility functions for various knowledge sources are described in section 5.4. In section 5.5, we conduct a simple experiment by modelling all-words WSD problem as a DCOP and perform disambiguation on Senseval-2 (Cotton et al., 2001) and Senseval-3 (Mihalcea and Edmonds, 2004) dataset of all-words task.

5.2 Distributed Constraint Optimization Problem (DCOP)

A DCOP (Modi, 2003; Modi et al., 2004) consists of n variables $V = x_1, x_2, ...x_n$ each assigned to an agent, where the values of the variables are taken from finite, discrete domains $D_1, D_2, ..., D_n$ respectively. Only the agent has knowledge and control over values assigned to variables associated to it. The goal for the agents is to choose values for variables such that a given global objective function is maximised. The objective function is described as the summation over a set of utility functions.

DCOP can be formalized as a tuple (A, V, D, C, F) where

- A = $\{a_1, a_2, \ldots a_n\}$ is a set of n agents,
- V = $\{x_1, x_2, \ldots x_n\}$ is a set of n variables, each one associated to an agent,
- D = $\{D_1, D_2, \ldots D_n\}$ is a set of finite and discrete domains each one associated to the corresponding variable,
- C = $\{f_k : D_i \times D_j \times \ldots D_m \rightarrow \Re\}$ is a set of constraints described by various utility functions f_k. The utility function f_k is defined over a subset of variables V. The domain of f_k represent the constraints C_{f_k} and $f_k(c)$ represents the utility associated with the constraint c, where $c \in C_{f_k}$.
- F = $\sum_k z_k \cdot f_k$ is the objective function to be maximised where z_k is the weight of the corresponding utility function f_k

An agent is allowed to communicate only with its neighbours. Agents communicate with each other to agree upon a solution which maximises the objective function.

5.3 WSD as a DCOP

Given a sequence of words W= $\{w_1, w_2, \ldots w_n\}$ with corresponding admissible senses $D_{w_i} = \{s^1_{w_i}, s^2_{w_i} \ldots\}$, we model WSD as DCOP as follows.

5.3.1 Agents

Each word w_i is treated as an agent. The agent (word) has knowledge and control of its values (senses).

5.3.2 Variables

Sense of a word varies and it is the one to be determined. We define the sense of a word as its variable. Each agent w_i is associated with the variable s_{w_i}. The value assigned to this variable indicates the sense assigned by the algorithm.

5.3.3 Domains

Senses of a word are finite in number. The set of senses D_{w_i}, is the domain of the variable s_{w_i}.

5.3.4 Constraints

A constraint specifies a particular configuration of the agents involved in its definition and has a utility associated with it. For e.g. If c_{ij} is a constraint defined on agents w_i and w_j, then c_{ij} refers to a particular instantiation of w_i and w_j, say $w_i = s^p_{w_i}$ and $w_j = s^q_{w_j}$.

A utility function $f_k : C_{f_k} \rightarrow \Re$ denote a set of constraints $C_{f_k} = \{D_{w_i} \times D_{w_j} \ldots D_{w_m}\}$, defined on the agents $w_i, w_j \ldots w_m$ and also the utilities associated with the constraints. We model information from each knowledge source as a utility function. In section 4, we decibel in detail about this modelling.

5.3.5 Objective function

As already stated, various knowledge sources are identified to be useful for WSD. It is desirable to use information from these sources collectively, to perform disambiguation. DCOP provides such framework where an objective function is defined over all the knowledge sources (f_k) as below

$$F = \sum_k z_k \cdot f_k$$

where F denotes the total utility associated with a solution and z_k is the weight given to a knowledge source i.e. information from various sources can be weighted. (Note: It is desirable to normalize utility functions of different knowledge sources in-order to compare them.)

Every agent (word) choose its value (sense) in a such a way that the objective function (global solution) is maximised. This way an agent is assigned a best value which is the target sense in our case.

5.4 Modelling information from various knowledge sources

In this section, we discuss the modelling of information from various knowledge sources.

5.4.1 Part-of-speech (POS)

Consider the word *play*. It has 47 senses out of which only 17 senses correspond to *noun* category. Based on the POS information of a word w_i, its domain D_{w_i} is restricted accordingly.

5.4.2 Morphology

Noun *orange* has atleast two senses, one corresponding to *a color* and other to *a fruit*. But plural form of this word *oranges* can only be used in the *fruit* sense. Depending upon the morphological information of a word w_i, its domain D_{w_i} can be restricted.

5.4.3 Domain information

In the sports domain, *cricket* likely refers to *a game* than *an insect*. Such information can be captured using a unary utility function defined for every word. If the sense distributions of a word w_i are known, a function $f : D_{w_i} \rightarrow \Re$ is defined which return higher utility for the senses favoured by the domain than to the other senses.

5.4.4 Sense Relatedness

Sense relatedness between senses of two words w_i, w_j is captured by a function $f : D_{w_i} \times D_{w_j} \rightarrow \Re$ where f returns sense relatedness (utility) between senses based on sense taxonomy and gloss overlaps.

5.4.5 Discourse

Discourse constraints can be modelled using a n-ary function. For instance, to the extent one sense per discourse (Gale et al., 1992) holds true, higher utility can be returned to the

solutions which favour same sense to all the occurrences of a word in a given discourse. This information can be modeled as follows: If $w_i, w_j, \ldots w_m$ are the occurrences of a same word, a function $f : D_i \times D_j \times \ldots D_m \rightarrow \Re$ is defined which returns higher utility when $s_{w_i} = s_{w_j} = \ldots s_{w_m}$ and for the rest of the combinations it returns lower utility.

5.4.6 Collocations

Collocations of a word are known to provide strong evidence for identifying correct sense of the word. For example: if in a given context *bank* co-occur with *money*, it is likely that *bank* refers to *financial institution* sense rather than *the edge of a river* sense. The word *cancer* has atleast two senses, one corresponding to the astrological sign and the other a disease. But its derived form *cancerous* can only be used in disease sense. When the words *cancer* and *cancerous* co-occur in a discourse, it is likely that the word *cancer* refers to *disease sense*.

Most supervised systems work through collocations to identify correct sense of a word. If a word w_i co-occurs with its collocate *v*, collocational information from v can be modeled by using the following function

$$coll\_infrm\_v_{w_i} : D_{w_i} \rightarrow \Re$$

where $coll\_infrm\_v_{w_i}$ returns high utility to collocationally preferred senses of w_i than other senses.

Collocations can also be modeled by assigning more than one variable to the agents or by adding a dummy agent which gives collocational information but in view of simplicity we do not go into those details.

Topical word associations, semantic word associations, selectional preferences can also be modeled similar to collocations. Complex information involving more than two entities can be modelled by using n-ary utility functions.

5.5 Experiment: DCOP based All Words WSD

We carried out a simple experiment to test the effectiveness of DCOP algorithm. We conducted our experiment in an all words setting and used only WordNet (Fellbaum, 1998)

based relatedness measures as knowledge source so that results can be compared with earlier state-of-art knowledge-based WSD systems like (Agirre and Soroa, 2009; Sinha and Mihalcea, 2007) which used similar knowledge sources as ours.

Our method performs disambiguation on sentence by sentence basis. A utility function based on semantic relatedness is defined for every pair of words falling in a particular window size. Restricting utility functions to a window size reduces the number of constraints. An objective function is defined as sum of these restricted utility functions over the entire sentence and thus allowing information flow across all the words. Hence, a DCOP algorithm which aims to maximize this objective function leads to a globally optimal solution.

In our experiments, we used the best similarity measure settings of (Sinha and Mihalcea, 2007) which is a sum of normalized similarity measures jcn, lch and lesk. We used used Distributed Pseudotree Optimization Procedure (DPOP) algorithm (Petcu and Faltings, 2005), which solves DCOP using linear number of messages among agents. The implementation provided with the open source toolkit FRODO[1] (Léauté et al., 2009) is used.

5.5.1 Data

To compare our results, we ran our experiments on SENSEVAL-2 (Cotton et al., 2001) and SENSEVAL -3 (Mihalcea and Edmonds, 2004) English all-words data sets.

5.5.2 Results

Table 5.1 shows results of our experiments. All these results are carried out using a window size of four. Ideally, precision and recall values are expected to be equal in our setting. But in certain cases, the tool we used, FRODO, failed to find a solution with the available memory resources.

Results show that our system performs consistently better than (Sinha and Mihalcea, 2007) which uses exactly same knowledge sources as used by us (with an exception of adverbs in Senseval-2). This shows that DCOP algorithm perform better than page-rank

[1] http://liawww.epfl.ch/frodo/

algorithm used in their graph based setting. Thus, for knowledge-based WSD, DCOP framework is a potential alternative to graph based models.

Table 1 also shows the system (Agirre and Soroa, 2009), which obtained best results for knowledge based WSD. A direct comparison between this and our system is not quantitative since they used additional knowledge such as extended WordNet relations (Mihalcea and Moldovan, 2001) and sense disambiguated gloss present in WordNet3.0.

Senseval-2 All Words data set					
	noun	verb	adj	adv	all
P_dcop	67.85	37.37	**62.72**	56.87	58.63
R_dcop	66.44	35.47	61.28	56.65	57.09
F_dcop	67.14	36.39	**61.99**	56.76	57.85
P_Sinha07	67.73	36.05	62.21	60.47	**58.83**
R_Sinha07	65.63	32.20	**61.42**	60.23	56.37
F_Sinha07	66.24	34.07	61.81	60.35	57.57
Agirre09	**70.40**	**38.90**	58.30	**70.1**	**58.6**
MFS	71.2	39.0	61.1	75.4	60.1
Senseval-3 All Words data set					
P_dcop	62.31	43.48	57.14	**100**	54.68
R_dcop	60.97	42.81	55.17	**100**	53.51
F_dcop	61.63	43.14	56.14	**100**	54.09
P_Sinha07	61.22	45.18	54.79	**100**	54.86
R_Sinha07	60.45	40.57	54.14	**100**	52.40
F_Sinha07	60.83	42.75	54.46	**100**	53.60
Agirre09	**64.1**	**46.9**	**62.6**	92.9	**57.4**
MFS	69.3	53.6	63.7	92.9	62.3

Table 5.1: Evaluation results on Senseval-2 and Senseval-3 data-set of all words task.

5.5.3 Performance analysis

We conducted our experiment on a computer with two 2.94 Ghz process and 2 GB memory. Our algorithm just took 5 minutes 31 seconds on senseval-2 data set, and 5 minutes 19 seconds on senseval-3 data set. This is a sizable reduction compared to execution time of page rank algorithms employed in both Sinha07 and Agirre09. In Agirre09, it falls in the range 30 to 180 minutes on much powerful system with 16 GB memory having four

2.66 Ghz processors. On our system, time taken by the page rank algorithm in (Sinha and Mihalcea, 2007) is 11 minutes when executed on senseval-2 data set.

Since DCOP algorithms are truly distributed in nature the execution times can be further reduced by running them parallely on multiple processors.

5.6 Related work

Earlier approaches to WSD which encoded information from variety of knowledge sources can be classified as follows:

- Supervised approaches: Most of the supervised systems (Yarowsky and Florian, 2002; Lee and Ng, 2002; Martínez et al., 2002; Stevenson and Wilks, 2001) rely on the sense tagged data. These are mainly discriminative or aggregative models which essentially pose WSD a classification problem. Discriminative models aim to identify the most informative feature and aggregative models make their decisions by combining all features. They disambiguate word by word and do not collectively disambiguate whole context and thereby do not capture all the relationships (e.g sense relatedness) among all the words. Further, they lack the ability to directly represent constraints like one sense per discourse.

- Graph based approaches: These approaches crucially rely on lexical knowledge base. Graph-based WSD approaches (Agirre and Soroa, 2009; Sinha and Mihalcea, 2007) perform disambiguation over a graph composed of senses (nodes) and relations between pairs of senses (edges). The edge weights encode information from a lexical knowledge base but lack an efficient way of modelling information from other knowledge sources like collocational information, selectional preferences, domain information, discourse. Also, the edges represent binary utility functions defined over two entities which lacks the ability to encode ternary, and in general, any N-ary utility functions.

5.7 Discussion

This framework provides a convenient way of integrating information from various knowledge sources by defining their utility functions. Information from different knowledge sources can be weighed based on the setting at hand. For example, in a domain specific WSD setting, sense distributions play a crucial role. The utility function corresponding to the sense distributions can be weighed higher in order to take advantage of domain information. Also, different combination of weights can be tried out for a given setting. Thus for a given WSD setting, this framework allows us to find 1) the impact of each knowledge source individually 2) the best combination of knowledge sources.

Limitations of DCOP algorithms: Solving DCOPs is NP-hard. A variety of search algorithms have therefore been developed to solve DCOPs (Mailler and Lesser, 2004; Modi et al., 2004; Petcu and Faltings, 2005) . As the number of constraints or words increase, the search space increases thereby increasing the time and memory bounds to solve them. Also DCOP algorithms exhibit a trade-off between memory used and number of messages communicated between agents. DPOP (Petcu and Faltings, 2005) use linear number of messages but requires exponential memory whereas ADOPT (Modi et al., 2004) exhibits linear memory complexity but exchange exponential number of messages. So it is crucial to choose a suitable algorithm based on the problem at hand.

5.8 Summary

In this chapter, we initiated a new line of investigation into WSD by modelling it in a distributed constraint optimization framework. We showed that this framework is powerful enough to encode information from various knowledge sources. Our experimental results show that a simple DCOP based model encoding just word similarity constraints performs comparably with the state-of-the-art knowledge based WSD systems. In our experiments, we only used relatedness based utility functions derived from WordNet. Effect of other knowledge sources remains to be evaluated individually and in combination. The best possible combination of weights of knowledge sources is yet to be engineered. Which DCOP algorithm performs better WSD and when has to be explored.

Chapter 6

Conclusions and Future Work

In this work we have introduced the problem of semantic category labeling (SCL), a coarse grained sense disambiguation problem. We have proposed two unsupervised corpus driven approaches for SCL - flat semantic category labeler (FSCL) and hierarchical semantic category labeler (HSCL). HSCL disambiguate in a top down manner and offers many advantages compared to the bottom up approaches like data sparsity, handling unseen instances and disambiguation with high precision values.

We have evaluated the usefulness of semantic relatedness measures for SCL and WSD tasks. It is found that disambiguation benefits from relatedness measures which can measure relatedness between concepts across different part-of-speech (POS) categories rather than the similarity measures which is in general between the same POS category. Lesk and adapted Lesk measures are found to be performing well for disambiguation task.

We have also evaluated the importance of domain knowledge in domain-specific WSD. Results show that domain play a key role and when domain information is incorporated with a normal method large performance drift is seen. It is also evident from the experiments that weakening the effect of non-domain senses gave us a good performance.

Above, we have seen the importance of each knowledge source individually. Ideally it is desirable to use many knowledge sources collectively and perform disambiguation arriving at a global optimal solution. In order to address this problem, we have proposed a distributed constraint framework for modelling information from various knowledge sources as constraints. Constraints are chosen such that a global optimum solution is attained

thereby performing WSD. Our initial experiments show that distributed constraint optimization framework is promising.

In the future, we would like to explore the effectiveness of HSCL for other languages. Furthermore, we would also like to test our domain-specific method on various domains. Currently, our DCOP framework has been tested only for semantic relatedness measures. In future, we aim to use more than one knowledge source in DCOP and also weigh each knowledge source based on its importance.

Appendix-1

Domain Specific Thesaurus, Sense Ranking Scores and Keyword Ranking Scores are accessible at

- `http://sivareddy.in/SemEval2010/thesauraus/`
- `http://sivareddy.in/SemEval2010/SenseRankingsUsingBgDocuments/`
- `http://sivareddy.in/SemEval2010/bgdocsKeyWords.txt.lemma.norm`

Tools Used:

- UKB is used with options *–ppr –dict_weight*. Dictionary files which UKB uses are automatically generated using sense ranking scores srs.
- Background document words are canonicalized using KSTEM, a morphological analyzer
- Stanford Parser is used to parse background documents to build thesaurus
- Test data is part of speech tagged using TreeTagger.

Bibliography

Agirre, E. and Martínez, D. (2001). Knowledge sources for word sense disambiguation. In *Text, Speech and Dialogue, 4th International Conference, TSD 2001, Zelezna Ruda, Czech Republic, September 11-13, 2001*, Lecture Notes in Computer Science. Springer, pages 1-10.

Agirre, E. and Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics, pages 33–41.

Agirre, E. and Stevenson, M. (2006). Knowledge sources for wsd. In *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*. Springer, pages 217–252, Dordrecht, The Netherlands.

Avinesh.PVS and Gali, K. (2007). Part-of-speech tagging and chunking using conditional random fields and transformation based learning. In *IJCAI-07 Workshop on "Shallow Parsing in South Asian Languages"*.

Banerjee, S. and Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, London, UK, pages 136–145. Springer-Verlag.

Bharati, A., Husain, S., Ambati, B., Jain, S., Sharma, D. M., and Sangal, R. (2008). Two semantic features make all the difference in parsing accuracy. In *International Conference*

on Natural Language Processing (ICON-08), CDAC, Pune, India. Macmillan Publishers India Ltd.

Buitelaar, P., Magnini, B., Strapparava, C., and Vossen, P. (2006). Domain-specific wsd. In *Word Sense Disambiguation. Algorithms and Applications, Editors: Eneko Agirre and Philip Edmonds*. Springer.

Carpuat, M. and Wu, D. (2007). Improving statistical machine translation using word sense disambiguation. In *In The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007*, pages 61–72.

Cotton, S., Edmonds, P., Kilgarriff, A., and Palmer, M. (2001). Senseval-2. `http://www.sle.sharp.co.uk/senseval2`.

Erk, K. and McCarthy, D. (2009). Graded word sense assignment. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 440–449, Morristown, NJ, USA. Association for Computational Linguistics.

Fellbaum, C., editor (1998). *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London.

Ferraresi, A., Zanchetta, E., Baroni, M., and Bernardini, S. (2008). Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceed-ings of the WAC4 Workshop at LREC 2008*, Marrakesh, Morocco.

Gale, W. A., Church, K. W., and Yarowsky, D. (1992). One sense per discourse. In *HLT '91: Proceedings of the workshop on Speech and Natural Language*, Morristown, NJ, USA. Association for Computational Linguistics, pages 233–237.

Hirst, G. (1987). *Semantic interpretation and the resolution of ambiguity*. Cambridge University Press, New York, NY, USA.

Ide, N. and Wilks, Y. (2006). Making Sense About Sense. In *Word Sense Disambiguation: Algorithms And Applications*, chapter 3. Springer, Dordrecht.

Jiang, J. and Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33.

Kavalec, M., Maedche, E., and Svtek, V. (2004). Discovery of lexical entries for non–taxonomic. In *Proceedings of SOFSEM 2004: Theory and Practice of Computer Science, LNCS 2932*, pages 249–256.

Kilgarriff, A. (1997). The hard parts of lexicography. *International Journal of Lexicography 11 (1)*.

Kilgarriff., A. (2001). English lexical sample task description. In *Proceedings of the SENSEVAL-2 workshop, ACL Workshop.*

Kilgarriff, A., Reddy, S., Pomiklek, J., and PVS, A. (2010). A corpus factory for many languages. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics, pages 423–430.

Koeling, R., McCarthy, D., and Carroll, J. (2005). Domain-specific sense distributions and predominant sense acquisition. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Morristown, NJ, USA. Association for Computational Linguistics, pages 419–426.

Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In *Fellbaum MIT Press*, pages 265–283.

Léauté, T., Ottens, B., and Szymanek, R. (2009). FRODO 2.0: An open-source framework for distributed constraint optimization. In *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*, Pasadena, California, USA, Pages 160–164.

Lee, Y. K. and Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *EMNLP '02: Proceedings of the ACL-02*

conference on Empirical methods in natural language processing, Morristown, NJ, USA. Association for Computational Linguistics, pages 41–48.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, New York, NY, USA. ACM, pages 24–26.

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of ACL/EACL-97*, pages 64–71.

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, Morristown, NJ, USA. Association for Computational Linguistics, pages 768–774.

Lin, D. (1998b). An information-theoretic definition of similarity. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, San Francisco, CA, USA, pages 296–304. Morgan Kaufmann Publishers Inc.

Mailler, R. and Lesser, V. (2004). Solving distributed constraint optimization problems using cooperative mediation. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, Washington, DC, USA. IEEE Computer Society, pages 438–445.

Màrquez, L., Exsudero, G., Martínez, D., and Rigau, G. (2006). Supervised corpus-based methods for wsd. In *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*, pages 167–216. Springer, Dordrecht, The Netherlands.

Martha Palmer, Christiane Fellbaum, S. C. L. D. and Dang., H. T. (2001). English tasks: All-words and verb lexical sample. In *Proceedings of the SENSEVAL -2 workshop, ACL Workshop*.

Martínez, D., Agirre, E., and Màrquez, L. (2002). Syntactic features for high precision word sense disambiguation. In *COLING*.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2004). Finding predominant word senses in untagged text. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics, pages 279.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2007). Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4).

McRoy, S. W. (1992). Using multiple knowledge sources for word sense discrimination. *COMPUTATIONAL LINGUISTICS*, 18:pages 1–30.

Mihalcea, R. (2006). Knowledge-based methods for wsd. In *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*, pages 107–132. Springer, Dordrecht, The Netherlands.

Mihalcea, R. and Edmonds, P., editors (2004). *Proceedings 3rd International Workshop on Evaluating Word Sense Disambiguation Systems.* ACL, Barcelona, Spain.

Mihalcea, R. and Moldovan, D. I. (2001). extended wordnet: progress report. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*, pages 95–100.

Miller, G. A., Leacock, C., Tengi, R., and Bunker, R. T. (1993). A semantic concordance. In *Proceedings of the ARPA Workshop on Human Language Technology*. Morgan Kaufman, pages 303–308.

Modi, P. J. (2003). Distributed constraint optimization for multiagent systems. *PhD Thesis*.

Modi, P. J., Shen, W.-M., Tambe, M., and Yokoo, M. (2004). Adopt: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161:149–180.

Narayan, D., Chakrabarti, D., Pande, P., and Bhattacharyya, P. (2002). An experience in building the indo wordnet-a wordnet for hindi. In *Proceedings of First International Conference on Global WordNet*.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

Patwardhan, S., Banerjee, S., and Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico, pages 241–257.

Pedersen, T. (2006). Unsupervised corpus-based methods for wsd. In *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*, pages 133–166. Springer, Dordrecht, The Netherlands.

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet::similarity: measuring the relatedness of concepts. In *HLT-NAACL '04: Demonstration Papers at HLT-NAACL 2004 on XX*, Morristown, NJ, USA. Association for Computational Linguistics, pages 38–41.

Petcu, A. and Faltings, B. (2005). A scalable method for multiagent constraint optimization. In *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pages 266–271.

Ramakrishnan, G., Prithviraj, B. P., Deepa, A., Bhattacharya, P., and Chakrabarti, S. (2004). Soft word sense disambiguation. In *The Second Global Wordnet Conference, Masaryk University Brno, Czech Republic*, pages 33–64.

Rayson, P. and Garside, R. (2000). Comparing corpora using frequency profiling. In *Proceedings of the workshop on Comparing corpora*, Morristown, NJ, USA. Association for Computational Linguistics, pages 1–6.

Resnik, P. (2006). Word sense disambiguation in natural language processing applications. In *Agirre E., & Edmonds, P. (Eds.), Word Sense Disambiguation, chap.11*, pages 299–337.

Sinha, M., Kumar, M., Pande, P., Kashyap, L., and Bhattacharyya, P. (2004). Hindi word sense disambiguation. In *International Symposium on Machine Translation, Natural Language Processing and Translation Support Systems, Delhi, India, November, 2004.*

Sinha, R. and Mihalcea, R. (2007). Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, Washington, DC, USA. IEEE Computer Society, pages 363–369.

Stevenson, M. and Wilks, Y. (2001). The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, 27(3).

Tugwell, D. and Kilgarriff, A. (2001). Wasp-bench: a lexicographic tool supportng word sense disambiguation. In *Proceedings of the SENSEVAL-2 Workshop. In conjunction with ACL-2001/EACL-2001, Toulouse, France*.

Vickrey, D., Biewald, L., Teyssier, M., and Koller, D. (2005). Word-sense disambiguation for machine translation. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Morristown, NJ, USA. Association for Computational Linguistics, pages 771–778.

Weeber, M., Weeber, M., Mork, J. G., and Aronson, A. R. (2001). Developing a test collection for biomedical word sense disambiguation. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA 2001). Philadelphia: Hanley & Belfus*, pages 746–750.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Morristown, NJ, USA. Association for Computational Linguistics, pages 133–138.

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings of COLING-92*, pages 454–460.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *ACL - 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, MIT, Cambridge, Massachusetts, USA. Morgan Kaufmann.

Yarowsky, D. and Florian, R. (2002). Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8.