

# Domain-Specific Word Sense Disambiguation combining corpus based and wordnet based parameters

Mitesh M. Khapra    Sapan Shah    Piyush Kedia    Pushpak Bhattacharyya

Department of Computer Science and Engineering

Indian Institute of Technology, Bombay

Powai, Mumbai – 400076,

Maharashtra, India.

{miteshk, sapan, charasi, pb}@cse.iitb.ac.in

## Abstract

We present here an algorithm for domain specific all-words WSD. The scoring function to rank the senses is inspired by the quadratic energy expression of Hopfield network, a well studied expression in neural networks. The scoring function is employed by a greedy iterative disambiguation algorithm that uses only the *words-disambiguated-so-far* to disambiguate the current word in focus. The combination of the algorithm and the scoring function seems to perform well in two ways: (i) the algorithm beats the domain corpus baseline which is typically hard to beat, and (ii) the algorithm is a good balance between efficiency and performance. The latter fact is established by comparing the iterative algorithm with a PageRank like disambiguation algorithm and an exhaustive sense graph search algorithm. The accuracy values of approximately 69% (F1-score) in two different domains- where the domain corpus baseline stands at 65%- compares very well with the state of the art.

## 1 Introduction

Sense distributions of words are highly skewed (Kilgarriff, 2004) and depend heavily on the domain (Magnini *et. al.*, 2002) at hand. This fact makes it very difficult for WSD approaches to *beat the corpus baseline*, as the common parlance goes. To disambiguate a word, simply pick the most frequent sense of that word in the corpus, *independent of the context*.

One could live with this situation, were the baseline performance good enough for most applications. But as an embedded module, *e.g.*, in a pipelined machine translation system, WSD should happen with very high precision and recall for lexical substitution to work effectively, and corpus baseline level performance is hardly adequate for this. For high accuracy disambigua-

tion it is imperative to accumulate and use the context evidence.

The difficulty of beating the corpus baseline was brought home by the task of evaluating a number of WSD systems for the English all-words task in SENSEVAL-3 (Snyder and Palmer, 2004). It was observed that only 5 out of the 26 systems were able to outperform the most frequent corpus sense heuristic derived from SemCor<sup>1</sup>.

Our work reported here, we admit, is on a beaten track. What is the need for yet another WSD algorithm? However, the demands of a large MT task described in the next para, coupled with the discussion on existing work will show that no final word has yet been said on the important problem of *all-word-domain-specific-WSD*, and the task will need all ingenuity and investment in methodology, tools and resources to obtain satisfactory solutions.

Large scale strictly result-oriented efforts are on in India to translate from English to Indian languages. The approach is essentially rule based, SMT being infeasible due to lack of large quantities of parallel corpora. WSD forms a critical component in this, influencing lexical substitution. The domains of interest are tourism and health and languages involved are *Hindi, Marathi, Punjabi, Bengali, Tamil, Kannada* and *Telugu*. The speaker population of each of these languages is hundreds of millions, with Hindi leading the pack at approximately 500 millions.

The organization of the paper is as follows: section 2 is on literature survey. Section 3 describes the parameters used in our scoring function for disambiguation. The description and the rationale behind the scoring function follow in section 4. Section 5 presents the three algorithms used by us for WSD, *viz.*, greedy and iterative, PageRank based and exhaustive search based. Section 6 discusses the results obtained. Section

---

<sup>1</sup> <http://multisemcor.itc.it/semcor.php>

7 gives a qualitative comparison of the three algorithms. Conclusions and future work are presented in section 8.

## 2 Literature survey

Knowledge based approaches to WSD such as Lesk's algorithm (Michael Lesk, 1986), Walker's algorithm (Walker D. & Amsler R., 1986), conceptual density (Agirre Eneko & German Rigau, 1996) and random walk algorithm (Mihalcea Rada, 2005) essentially do Machine Readable Dictionary lookup. However, these are fundamentally *overlap based* algorithms which suffer from overlap sparsity, dictionary definitions being generally small in length. Further, these algorithms completely ignore the domain specific sense distributions of a word as they do not rely on any training data.

Supervised learning algorithms for WSD are mostly word specific classifiers, *e.g.*, WSD using SVM (Lee et al., 2004), Exemplar based WSD (Ng Hwee T. & Hian B. Lee., 1996) and decision list based algorithm (Yarowsky, 1994). To the best of our knowledge none of these algorithms have been adapted to the task of domain-specific all-words disambiguation.

Semi-supervised and unsupervised algorithms do not need large amount of annotated corpora, but are again word specific classifiers, *e.g.*, semi-supervised decision list algorithm (Yarowsky, 1995) and Hyperlex (Véronis Jean, 2004). Hybrid approaches like WSD using Structural Semantic Interconnections (Roberto Navigli & Paolo Velardi, 2005) use combinations of more than one knowledge sources (wordnet as well as a small amount of tagged corpora). This allows them to capture important information encoded in wordnet (Fellbaum, 1998) as well as draw syntactic generalizations from minimally tagged corpora. *These methods which combine evidence from several resources seem to be most suitable in building all-words disambiguation engines and are the motivation for our work.*

Previous attempts at domain specific WSD have emphasized the correlation between domain and sense distributions (Magnini *et. al.*, 2002) and have focused on learning the distributions of a small set of high frequency words in an unsupervised (Agirre *et. al.*, 2009) or supervised manner (Koeling *et. al.*, 2005; Agirre and Lopez, 2008; Agirre *et. al.*, 2009). In this paper we emphasize the importance of other factors dependent on the sentential context and show that

combining these with the domain specific sense distributions can help to beat corpus baseline.

## 3 Parameters essential for domain-specific WSD

We discuss a number of parameters that play a crucial role in WSD. To appreciate this, consider the following example:

*The river flows through this region to meet the sea.*

The word *sea* is ambiguous and has three senses as given in the Princeton Wordnet (PWN):

*S1: (n) sea (a division of an ocean or a large body of salt water partially enclosed by land)*

*S2: (n) ocean, sea (anything apparently limitless in quantity or volume)*

*S3: (n) sea (turbulent water with swells of considerable size) "heavy seas"*

Our first parameter is obtained from **Domain specific sense distributions**. In the above example, the first sense is more frequent in the tourism domain (verified from manually sense marked tourism corpora). Domain specific sense distribution information should be harnessed in the WSD task.

The second parameter arises from the **dominance of senses in the domain**. Senses are expressed by synsets, and we define a dominant sense as follows:

A synset node in the wordnet hypernymy hierarchy is called *Dominant* if the synsets in the sub-tree below the synset are frequently occurring in the domain corpora.

A few dominant senses in the Tourism domain are *{place, country, city, area}*, *{body of water}*, *{flora, fauna}*, *{mode of transport}* and *{fine arts}*. In disambiguating a word, that sense which belongs to the sub-tree of a domain-specific dominant sense should be given a higher score than other senses. The value of this parameter ( $\theta$ ) is decided as follows:

$\theta = 1$ ; if the candidate synset is a dominant synset

$\theta = 0.5$ ; if the candidate synset belongs to the sub-tree of a dominant synset

$\theta = 0.001$ ; if the candidate synset is neither a dominant synset nor belongs to the sub-tree of a dominant synset.

Our third parameter comes from **Corpus co-occurrence**. Co-occurring monosemous words as well as *already disambiguated words* in the context help in disambiguation. For example, the word *river* appearing in the context of *sea* is a monosemous word. The frequency of co-occurrence of *river* with the “water body” sense of *sea* is high in the tourism domain.

Our fourth parameter is based on the **semantic distance** between any pair of synsets in terms of the shortest path length between two synsets in the wordnet graph. An edge in the shortest path can be any semantic relation from the wordnet relation repository (e.g., *hypernymy*, *hyponymy*, *meronymy*, *holonymy*, *troponymy* etc.).

For nouns we do something additional over and above the semantic distance. We take advantage of the deeper hierarchy of noun senses in the wordnet structure. This gives rise to our fifth and final parameter which arises out of the **conceptual distance** between a pair of senses. Conceptual distance between two synsets  $S_1$  and  $S_2$  is calculated using Equation (1), motivated by Agirre Eneko & German Rigau (1996).

$$\text{Conceptual Distance } (S_1, S_2) = \frac{\text{Length of the path between } (S_1, S_2) \text{ in terms of hypernymy hierarchy}}{\text{Height of the lowest common ancestor of } S_1 \text{ and } S_2 \text{ in the wordnet hierarchy}} \quad (1)$$

The conceptual distance is proportional to the path length between the synsets, as it should be. The distance is also inversely proportional to the height of the common ancestor of two sense nodes, because as the common ancestor becomes more and more general the conceptual relatedness tends to get vacuous (e.g., two nodes being related through *entity* which is the common ancestor of EVERYTHING, does not really say anything about the relatedness).

To summarize, our various parameters used for domain-specific WSD are:

#### Wordnet-dependent parameters

- *belongingness-to-dominant-concept*
- *conceptual-distance*
- *semantic-distance*

#### Corpus-dependent parameters

- *sense distributions*
- *corpus co-occurrences*.

## 4 Our scoring function

We desired a scoring function which:

- (1) Uses the strong clues for disambiguation provided by the monosemous words and also the already disambiguated words.
- (2) Uses sense distributions learnt from a sense tagged corpus.
- (3) Captures the effect of dominant concepts within a domain.
- (4) Captures the interaction of a candidate synset with others synsets in the sentence.

We have been motivated by the Energy expression in Hopfield network (Hopfield, 1982) in formulating a scoring function for ranking the senses. Hopfield Network is a fully connected bidirectional symmetric network of bi-polar (0/1 or +1/-1) neurons. We consider the asynchronous Hopfield Network. At any instant, a randomly chosen neuron (a) examines the weighted sum of the input, (b) compares this value with a threshold and (c) gets to the state of 1 or 0, depending on whether the input is greater than or less than or equal to the threshold. The assembly of 0/1 states of individual neurons defines a state of the whole network. Each state has associated with it an energy,  $E$ , given by the following expression

$$E = \theta_i V_i - \sum_{i=1}^N \sum_{j>i}^N W_{ij} V_i V_j \quad (2)$$

where,  $N$  is the total number of neurons in the network,  $V_i$  and  $V_j$  are the activations of neurons  $i$  and  $j$  respectively and  $W_{ij}$  is the weight of the connection between neurons  $i$  and  $j$ . Energy is a fundamental property of Hopfield networks, providing the necessary machinery for discussing convergence, stability and such other considerations.

The energy expression as given above cleanly separates the influence of self-activations of neurons and that of interactions amongst neurons to the global macroscopic property of energy of the network. This fact has been the primary insight for equation (3) which was proposed to score the most appropriate synset in the given context. The correspondences are as follows:

Neuron	→	Synset
Self-activation	→	Corpus Sense Distribution
		Weight as a function of

Weight of connection between two neurons  $\rightarrow$  corpus co-occurrence and Wordnet distance measures between synsets

$$S^* = \underset{i}{\operatorname{argmax}} \left( \theta_i * V_i + \sum_{j \in J} W_{ij} * V_i * V_j \right) \quad (3)$$

where,

$J = \text{Set of disambiguated Words}$

$\theta_i = \text{BelongingnessToDominantConcept}(S_i)$

$V_i = P(S_i | \text{word})$

$W_{ij} = \text{CorpusCooccurrences}(S_i, S_j)$

$* 1/\text{WNConceptualDistance}(S_i, S_j)$

$* 1/\text{WNSemanticGraphDistance}(S_i, S_j)$

The component  $\theta_i * V_i$  is the energy due to the self activation of a neuron and can be compared to the corpus specific sense of a word in a domain. The other component  $w_{ij} * V_j * V_i$  coming from the interaction of activations can be compared to the score of a sense due to its interaction in the form of corpus co-occurrence, conceptual distance, and wordnet-based semantic distance with other words in the sentence. The first component thus captures the rather *static corpus sense*, whereas the second expression brings in the sentential context.

## 5 Our algorithms for WSD

We present three algorithms which combine the parameters described above to arrive at sense decisions, *viz.*, (i) a greedy iterative algorithm (ii) an exhaustive graph search algorithm and (iii) a modified PageRank algorithm.

### 5.1 Algorithm-1: Iterative WSD (IWSD)

---

**Algorithm 1:** *performIterativeWSD(sentence)*

---

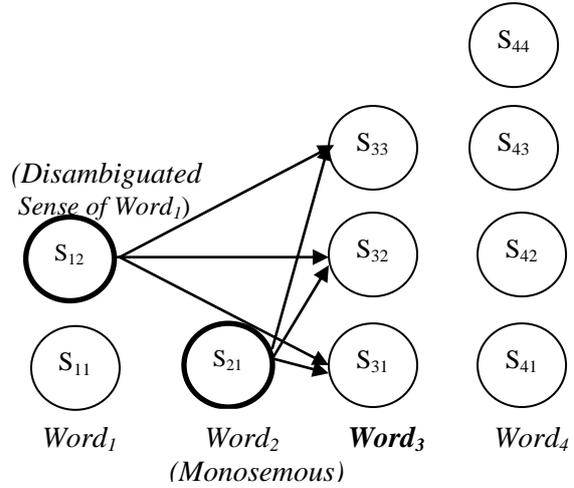
1. Tag all monosemous words in the sentence.
  2. Iteratively disambiguate the remaining words in the sentence in increasing order of their degree of polysemy.
  3. At each stage select that sense for a word which maximizes the score given by Equation (3)
- 

**Algorithm1:** Iterative WSD

Monosemous words are used as the seed input for the algorithm. Note that they are left out of consideration while calculating the precision and recall values. In case there are no monosemous words in the sentence, the disambiguation will be started with the first term in the formula which represents the corpus bias (the second term will

not be active as there are no previously disambiguated words). The least polysemous word thus disambiguated will then act as the seed input to the algorithm.

IWSD is clearly greedy. It bases its decisions on already disambiguated words, and ignores completely words with higher degree of polysemy. As shown in Figure 1, Word<sub>3</sub> is the current polysemous word being disambiguated. The algorithm only considers the interaction of its candidate senses with previously disambiguated and monosemous words in the context (shown in dark circles). Word<sub>4</sub> (which is more polysemous than Word<sub>3</sub>) does not come into picture.



**Figure 1:** Greedy operation of IWSD.

### 5.2 Algorithm-2: Exhaustive graph search algorithm

Suppose there are  $n$  words  $W_1, W_2, W_3, \dots, W_n$  in a sentence with  $m_1, m_2, m_3, \dots, m_n$  senses. WSD can then be viewed as the task of finding the best possible combination of senses from the possible  $m_1 * m_2 * m_3 * \dots * m_n$  combinations.

Each of these combinations can be assigned a score, and the combination with the highest score gets selected. The score of each node in the combination can be calculated using Equation (4).

$$\begin{aligned} \text{score}(\text{node}_i) &= \theta_i * V_i + \sum_{\substack{j \in \text{all Words} \\ j \neq i}} W_{ij} * V_i * V_j \quad (4) \end{aligned}$$

The terms on the RHS have the same meaning as in equation (3). Note that the summation in the second term is performed over all words as opposed to IWSD where the summation was performed only over previously disambiguated words. Thus unlike IWSD, this algorithm allows

all the words, already disambiguated or otherwise, to influence the decision for the current polysemous word.

The score of a combination is simply the sum of the scores of the individual nodes in the combination.

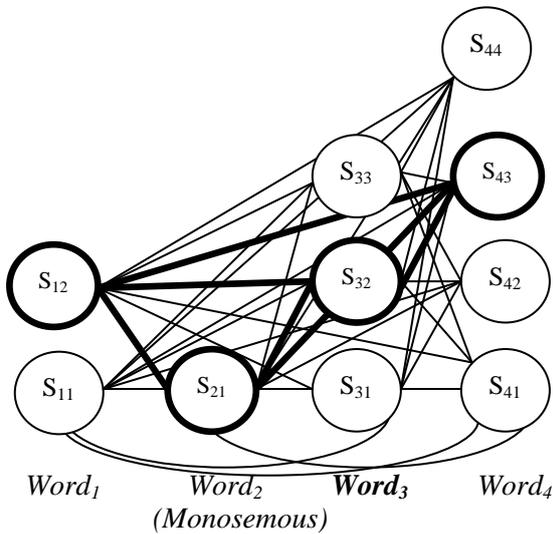
$score(combination)$

$$= \sum_{j \in C} \theta_j * V_j + 0.5 * \sum_{j \in C} \sum_{\substack{i \in C \\ i \neq j}} W_{ij} * V_i * V_j \quad (5)$$

where,

$C = \text{all words in the context}$

Note: The second term is multiplied by half to account for the fact that each term in the summation is added twice.



**Figure 2:** Exhaustive operation of graph search method.

As shown in Figure 2, there is an edge between every sense of every word and every sense of every other word which means that every word influences the sense decision for every other word. Contrast this with IWSD where Word<sub>4</sub> had no say in the disambiguation of Word<sub>3</sub>. Also the objective here is to select the best combination at one go as compared to IWSD which disambiguates only one word at a time. Note that each combination **must** contain at most and at least one sense node corresponding to every word. A possible best combination along with the connecting edges is highlighted in Figure 2.

This is definitely not a practical approach as it searches all the possible  $m_1 * m_2 * m_3 \dots * m_n$  combinations to find the best combination and hence has exponential time complexity. However, we still present it for the purpose of comparison.

### 5.3 Modifying PageRank to handle domain-specificity

Rada Mihalcea (2005) proposed the idea of using PageRank algorithm to find the best combination of senses in a sense graph. PageRank is a Random Walk algorithm used to find the importance of a vertex in a graph. It uses the idea of voting or recommendation. When one vertex links to another vertex it is basically casting a vote for that vertex (something like “This synset is semantically related to me, hence I am linking to it”). The nodes in a sense graph correspond to the senses of all the words in a sentence and the edges depict the strength of interaction between senses. The score of each node in the graph is then calculated using the following recursive formula:

$$score(S_i) = (1 - d) + d * \sum_{S_j \in In(S_i)} \frac{W_{ij}}{\sum_{S_k \in Out(S_j)} W_{jk}} * Score(S_j)$$

Instead of calculating  $W_{ij}$  based on the overlap between the definition of senses  $S_i$  and  $S_j$  as proposed by Rada Mihalcea (2005), we calculate the edge weights using the following formula:

$$W_{ij} = CorpusCooccurrences(S_i, S_j) * 1/WNConceptualDistance(S_i, S_j) * 1/WNSemanticGraphDistance(S_i, S_j) * P(S_i | word_x) * P(S_j | word_y) * \theta_i$$

$d = \text{damping factor (typically 0.85)}$

This formula helps capture the edge weights in terms of the corpus bias as well as the interaction between the senses in the corpus and wordnet. Just like exhaustive graph search, PageRank allows every word to influence the sense decision for every other word. Also the algorithm aims to select the overall best *combination* in the graph as opposed to IWSD where the aim is to disambiguate one word at a time. Even though PageRank and the graph search method look similar there is a subtle difference in the scoring functions used. PageRank uses a recursive scoring function, where the score of every node is updated in every iteration, whereas the graph search method uses a static formula which calculates the score of each node only once. Following Rada Mihalcea (2005), even we set the value of  $d$  as 0.85

Algorithm	Tourism Domain			Health Domain		
	P %	R %	F %	P %	R %	F %
Iterative WSD	72.08	67.33	<b>69.67</b>	78.74	72.15	<b>75.30</b>
PageRank	65.56	65.56	65.56	71.26	71.26	71.26
Wordnet Baseline	61.50	61.50	61.50	66.55	66.55	66.55
Corpus Baseline	73.60	58.41	65.13	81.06	63.92	71.48

**Table 3:** Precision, Recall and F-scores of IWSD, PageRank, Wordnet Baseline and Corpus Baseline

## 6 Results

We tested our algorithm on tourism and health corpus for English. The corpus was manually sense tagged by lexicographers using Princeton Wordnet 2.1 as the sense repository. We report our results on polysemous words only, *i.e.*, words which have at least one sense listed in the Wordnet. As mentioned earlier, monosemous words are used as the seed input for the algorithm but they are not considered while calculating the precision and recall values. The number of polysemous tokens and the average degree of polysemy in each domain is as mentioned in Table 1.

Domain	# of polysemous tokens	Average degree of polysemy
<i>Tourism</i>	32715	5.62
<i>Health</i>	14508	3.74

**Table 1:** Corpus size for each domain

We present the following results:

- (i) effectiveness of the proposed scoring function (*section 6.1*)
- (ii) comparison of IWSD with PageRank and corpus baseline for two domains (*Health and Tourism*) (*section 6.2*)
- (iii) comparison of greedy IWSD with exhaustive graph search method (*section 6.3*)

### 6.1 Effectiveness of the scoring function: does it represent the training data?

An oft-repeated question in machine learning is: *does the hypothesis learnt at least represent the training data?* Since the scoring function in Equation (2) was arrived at rather intuitively, taking clues from Hopfield network, we wanted to see if it actually represents the training data faithfully. For this, we removed the existing sense labels of the training data and relabeled them using our scoring function by running IWSD described in

section 5. Table 2 compares the performance of IWSD and corpus baseline on the training data.

Domain	Algorithm	P %	R %	F %
Tourism	IWSD	84.10	84.10	<b>84.10</b>
	Corpus Baseline	81.83	72.93	77.12
Health	IWSD	89.02	89.02	<b>89.02</b>
	Corpus Baseline	87.12	78.43	82.54

**Table 2:** Results on Training Data

F-scores of 84% and 89% on the two domains show that the proposed scoring function not only fits the training data well but also performs better than the corpus baseline.

### 6.2 Performance on test data

A 4-fold cross validation was done in both the domains. The results of Iterative WSD were compared with PageRank, wordnet baseline (*i.e.*, selecting the first sense from wordnet) as well as corpus baseline (*i.e.*, selecting the most frequent sense from the corpus). The results for are summarized in Table 3. We report both precision and recall values.

We observe that:

1. *IWSD performs better than wordnet baseline:* this is expected since the wordnet sense order does not represent the domain specific corpus sense distribution. Note that here recall is different from coverage. For example, for the Wordnet baseline the coverage would be 100% as every test word has a sense listed in the Wordnet and hence the engine can output the first sense for every test word. However, the recall will be low (61.5%) as recall measures the percentage of test words that were labeled correctly. Hence, in the case of Wordnet baseline, recall is the same as precision.
2. *IWSD performs better than corpus baseline:* IWSD beats the corpus baseline by 4.54% (F-score) in the Tourism domain and around

3.82% (F-score) in the Health domain. This once again establishes the soundness of the proposed scoring function as it shows that combining the self energy and the interaction energy indeed boosts the performance. We also note that in both the domains the precision of most frequent corpus sense is higher than IWSD but the recall is lower than IWSD. This reiterates the fact that domain-specific sense distributions *when available* are pretty accurate (high value of precision), but they may not be available for all words in the test corpus (low value of recall). For such cases where the domain-specific sense distribution is not available the only hope of disambiguation is through the *interaction energy* with neighboring senses.

3. *IWSD performs better than PageRank*: Both IWSD and PageRank make use of the self energy of the node as well as the context dependent energy arising from interactions with the neighboring senses. However, whereas IWSD does better than the corpus baseline in both the domains, PageRank performs only slightly better than the corpus baseline (+0.43%) in the Tourism domain and performs poorly as compared to corpus baseline in the Health domain (-0.22%). The better performance of IWSD over PageRank ( $\approx 4\%$  in both the domains) shows that the scoring function based on Hopfield network is a better way of combining energies than the iterative formula of PageRank.

### 6.3 Greedy v/s Exhaustive

Since the exhaustive sense graph search method is exponential in nature we could run it only on a small fraction (1%) of the test data in each fold. The results were compared with greedy IWSD and are summarized in Table 4:

We observe that the exhaustive method performs better than the greedy method in both the domains (F-scores: +1.3% for Tourism and +7.14% for Health). However, the exponential nature of the exhaustive graph search algorithm renders it useless for practical purposes (*e.g.*, even to run on only 1% of the test data the exhaustive search method takes 2 hours whereas IWSD takes only 1 minute). IWSD thus emerges as a practical alternative. But a question which is still left unanswered and which we intend to explore in our future work is whether we can use other graph search algorithms like Beam Search to close the performance gap (around 7.14%) between IWSD

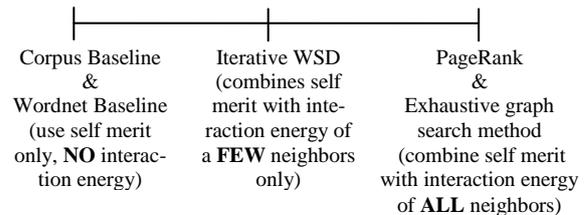
and exhaustive graph search method with some increase in the computational complexity.

Domain	Algorithm	P %	R %	F %
Tourism	IWSD	85.34	84.93	85.13
	Exhaustive graph search method	86.42	86.42	<b>86.42</b>
Health	IWSD	82.00	62.26	70.78
	Exhaustive graph search method	77.82	77.82	<b>77.82</b>

**Table 4:** Precision, Recall and F-scores of IWSD and exhaustive graph search method on a small fraction (1%) of the data for both the domains.

## 7 A qualitative comparison of the three algorithms presented

After the above exposition, we would first like to give an intuitive and qualitative comparison of the three algorithms we have seen. Corpus baseline and Wordnet baseline lie at one end of the spectrum as they rely only on the self energy of the node (in terms of ranking in corpus and ranking in wordnet respectively) and completely ignore the interaction with other senses in the context. PageRank and exhaustive graph search method lie at the other end of the spectrum as they combine the self energy with the interaction energy derived from the interaction with ALL words in the context. However, both these algorithms fail to strike a balance between performance and implementation feasibility. PageRank has implementation feasibility but lacks performance whereas the exhaustive graph search method gives better performance but lacks implementation feasibility. IWSD lies somewhere at the middle of the spectrum as it combines the self energy of a node with its interaction energy based on interaction with only FEW (previously disambiguated) words in the context. By doing so it is able to strike a balance between performance and implementation feasibility.



**Figure 3:** A spectrum showing the position of different WSD algorithms.

## 8 Conclusion and Future Work:

Based on our study for 2 domains, we conclude the following:

- (i) domain-specific sense distributions - if obtainable - can be exploited to advantage.
- (ii) combining self energy with interaction energy gives better results than using only self energy.
- (iii) making greedy local decisions and restricting more polysemous words from influencing the decision for less polysemous words works sufficiently well for domain-specific WSD.
- (iv) IWSD is able to strike a perfect balance between performance and implementation feasibility. None of the other algorithms are able to achieve this balance.

It would be interesting to test our algorithm on other domains and other languages to conclusively establish the significance of the proposed scoring function for WSD. It would also be interesting to check the domain-dependence of our algorithm by testing it on the SENSEVAL-3 dataset which contains general data not specific to any domain.

The exhaustive graph search method gives improvement in performance over IWSD but is computationally infeasible. It would be worth exploring other graph search methods like beam search which are computationally feasible and might perform somewhere in between IWSD and exhaustive graph search.

## References

- Adam Kilgarriff. 2004. *How dominant is the commonest sense of a word?* In Proceedings of Text, Speech, Dialogue, Brno, Czech Republic.
- Agirre Eneko & German Rigau. 1996. *Word sense disambiguation using conceptual density*. In Proceedings of the 16th International Conference on Computational Linguistics (COLING), Copenhagen, Denmark.
- Agirre, E. and Lopez de Lacalle, O. On Robustness and Domain Adaptation using SVD for Word Sense Disambiguation. COLING-08
- Agirre, E. and Lopez de Lacalle, O. Supervised Domain Adaption for WSD. EACL-09
- Agirre, E., Lopez de Lacalle, O. and Soroa, A. Knowledge-based WSD on Specific Domains: Performing better than Generic Supervised WSD. IJCAI-09
- Benjamin Snyder and Martha Palmer. 2004. *The English all-words task*. In Proceedings of SENSEVAL-3, pages 41–43, Barcelona, Spain.
- Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. 2002. *The role of domain information in word sense disambiguation*. Natural Language Engineering, 8(4):359–373.tx
- Dipak Narayan, Debasri Chakrabarti, Prabhakar Pande and P. Bhattacharyya. 2002. *An Experience in Building the Indo WordNet - a WordNet for Hindi*. First International Conference on Global WordNet, Mysore, India.
- Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.
- English Wordnet.  
<http://wordnet.princeton.edu/perl/webwn?s=word-you-want>
- J. J. Hopfield. April 1982. "Neural networks and physical systems with emergent collective computational abilities", Proceedings of the National Academy of Sciences of the USA, vol. 79 no. 8 pp. 2554-2558.
- Lee Yoong K., Hwee T. Ng & Tee K. Chia. 2004. *Supervised word sense disambiguation with support vector machines and multiple knowledge sources*. Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, 137-140.
- Lin Dekang. 1997. *Using syntactic dependency as local context to resolve word sense ambiguity*. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL), Madrid, 64-71.
- Michael Lesk. 1986. *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone*. In Proceedings of the 5th annual international conference on Systems documentation, Toronto, Ontario, Canada.
- Mihalcea Rada. 2005. *Large vocabulary unsupervised word sense disambiguation with graph-based algorithms for sequence data labeling*. In Proceedings of the Joint Human Language Technology and Empirical Methods in Natural Language Processing Conference (HLT/EMNLP), Vancouver, Canada, 411-418.
- Ng Hwee T. & Hian B. Lee. 1996. *Integrating multiple knowledge sources to disambiguate word senses: An exemplar-based approach*. In Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL), Santa Cruz, U.S.A., 40-47.
- Rajat Mohanty, Pushpak Bhattacharyya, Prabhakar Pande, Shraddha Kalele, Mitesh Khapra and Aditya Sharma. 2008. *Synset Based Multilingual*

- Dictionary: Insights, Applications and Challenges*. Global Wordnet Conference, Szeged, Hungary, January 22-25.
- Resnik Philip. 1997. *Selectional preference and sense disambiguation*. In Proceedings of ACL Workshop on Tagging Text with Lexical Semantics, Why, What and How? Washington, U.S.A., 52-57.
- Rob Koeling, Diana McCarthy, John Carroll. 2005. *Domain-specific sense distributions and predominant sense acquisition*, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, p.419-426, Vancouver, British Columbia, Canada.
- Roberto Navigli, Paolo Velardi. 2005. *Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation*. IEEE Transactions On Pattern Analysis and Machine Intelligence.
- Véronis Jean. 2004. *HyperLex: Lexical cartography for information retrieval*. Computer Speech & Language, 18(3):223-252.
- Walker D. and Amsler R. 1986. *The Use of Machine Readable Dictionaries in Sublanguage Analysis*. In Analyzing Language in Restricted Domains, Grishman and Kittredge (eds), LEA Press, pp. 69-83.
- Yarowsky David. 1994. *Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French*. In Proceedings of the 32nd Annual Meeting of the association for Computational Linguistics (ACL), Las Cruces, U.S.A., 88-95.
- Yarowsky David. 1995. *Unsupervised word sense disambiguation rivaling supervised methods*. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL), Cambridge, MA, 189-196.