Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Unsupervised Word Sense Disambiguation Using Markov Random Field and Dependency Parser

Devendra Singh Chaplot
Samsung Electronics Co., Ltd.
Suwon, South Korea

Pushpak Bhattacharyya
Computer Science Department
IIT Bombay, India

Ashwin Paranjape
Stanford University
California, US

Abstract

Word Sense Disambiguation is a difficult problem to solve in the unsupervised setting. This is because in this setting inference becomes more dependent on the interplay between different senses in the context due to unavailability of learning resources. Using two basic ideas, sense dependency and selective dependency, we model the WSD problem as a Maximum A Posteriori (MAP) Inference Query on a Markov Random Field (MRF) built using WordNet and Link Parser or Stanford Parser. To the best of our knowledge this combination of dependency and MRF is novel, and our graph-based unsupervised WSD system beats state-of-the-art system on SensEval-2, SensEval-3 and SemEval-2007 English all-words datasets while being over 35 times faster.

1 Introduction

Word Sense Disambiguation (WSD) is an open problem, concerned with understanding the sense of a word when the word has multiple meanings. The problem is important as it is useful in many other Natural Language Processing (NLP) tasks like machine translation (Chan, Ng, and Chiang 2007), information extraction (Zhong and Ng 2012) and question answering (Ramakrishnan et al. 2003). Due to the difficulty in creating quality sense annotated corpora in adequate amount, there has always been significant interest in developing unsupervised WSD algorithms.

We have built an entirely unsupervised WSD system which requires WordNet (Miller 1995), a dependency parser and Stanford POS Tagger (Toutanova et al. 2003) as knowledge resources. We model the WSD problem as a Maximum A Posteriori (MAP) Inference Query on a Markov Random Field (MRF) (Jordan and Weiss 2002), which is an undirected graphical model. Senses of words in the sentence form the nodes of this graphical model while the edges are determined using a dependency parser. We present our results using two different dependency parsers, Link Parser (Sleator and Temperley 1993) and Stanford Parser (Marneffe, Maccartney, and Manning 2006).

The rest of the paper is divided into the following sections. Section 2 covers Related Work regarding knowledge-based and unsupervised WSD and previous attempts using graph-based algorithms. Section 3 describes the methodology behind our unsupervised approach. Our algorithm is described in detail in Section 4. The thought process behind the development of our algorithm is discussed in Section 5. The experiments and results are presented in Section 6. Efficiency and labelling speed of the proposed system is discussed in Section 7. Section 8 describes the web interface developed for experiments, visualization, error analysis and optimization which shall be made open-source. Conclusions and future work are covered in Section 9.

2 Related Work

In recent times, many graph-based algorithms have been proposed for unsupervised WSD. Navigli and Lapata (2007) and Navigli and Lapata (2010) build a subgraph of the entire lexicon containing vertices useful for disambiguation and then use graph connectivity measures to determine the most appropriate senses. Mihalcea (2005) and Sinha and Mihalcea (2007) construct a sentence-wise graph, where, for each word every possible sense forms a vertex. Then graph-based iterative ranking and centrality algorithms are applied to find most probable sense. Agirre, López de Lacalle, and Soroa (2014) use personalised page rank over the graphs generated using WordNet.

A common recurrent problem in graph-based WSD algorithms is exponential complexity due to pairwise comparison of senses of all the words in the sentence. The search space for a sentence becomes the product of total number of possible senses of each content word. Due to this exponential search space, many sub-optimal and approximate techniques have been used. Patwardhan, Banerjee, and Pedersen (2003) use sub-optimal word-by-word greedy technique, while approximate techniques such as Simulated Annealing (Cowie, Guthrie, and Guthrie 1992), Conceptual Density (Agirre and Rigau 1996) and approximate solutions of equivalent problems in integer linear programming (Panagiotopoulou et al. 2012) have been tried out. Our algorithm reduces the search space by reducing the number of edges in the graphical model using a dependency parser, which allows us to exactly calculate the optimal solution unlike the above methods, and thereby increasing the accuracy of the system.

3 Basic Methodology

The proposed algorithm is based on two basic ideas.

- **Sense dependency:** Sense of a word depends on sense of other words in the sentence, not the words themselves.
- **Selective dependency:** Sense of a word depends on sense of only few other words in the sentence, not all.

Sense dependency

We argue that sense of a word should depend on sense of other words in the sentence, not the words themselves because it can be misguided by other words in the sentence. For example, consider the sentence,

"Getting rid of crickets is no game."

Here, sense of word 'game' is

game (frivolous or trifling behaviour) *"for actors, memorizing lines is no game"; "for him, life is all fun and games"*

If we use words in the sentence to determine the sense of a word, the word 'game' will misguide or drift the sense of the word 'cricket' towards

cricket (a game played with a ball and bat by two teams of 11 players; teams take turns trying to score runs)

while its correct sense would be

cricket (leaping insect; male, makes chirping noises by rubbing the forewings together).

If we use sense of other words to determine the sense of a word, this sense drift shouldn't occur.

Since we do not know the sense of any word in the sentence, we need to maximize the joint probability of senses of all the words in the sentence or in other words, probability of sense of the sentence rather than probability of senses of individual words.

Selective dependency

The sense of a word depends on (senses of) other words in its 'context'. The 'context' of a word is defined to be

context, linguistic context, context of use (discourse that surrounds a language unit and helps to determine its interpretation)

It is very difficult to determine the context of any given word. Traditionally, in word sense disambiguation techniques, the sentence in which the word occurs is considered to be its context. However, sentences are usually long, have many clauses and may contain words which can cause sense drift. For example, consider the sentence,

"They were troubled by insects while playing cricket."

Here, the sense of word 'insect' is

insect (small air-breathing arthropod)

This will misguide or drift the sense of word 'cricket' towards the sense of 'leaping insect', while its correct sense would be 'game played with bat and ball'.

Therefore, sense of a word depends on sense of only few other words in the sentence, not all. We argue that the context of a word depends on the syntactic structure of the sentence. We use a dependency parser to determine context of all words in the given sentence.

4 Proposed Algorithm

Using both the ideas from the previous section, we conclude that the sense of a word depends on senses of few other words in the sentence. Our goal is to maximize the joint probability of senses of all the words in the sentence, given sense dependencies of each word. These sense dependencies are determined using a dependency parser, while the required joint probability is maximized using Maximum A Posteriori (MAP) Inference Query on the Markov Random Field (MRF) constructed using the dependency parser and WordNet. Figure 1 shows a block diagram describing the proposed algorithm. It can be broadly divided into two parts:-

1. Construction of Markov Random Field (MRF)
2. Maximizing the joint probability using MAP Inference Query

Construction of Markov Random Field (MRF)

Construction of Markov Random Field requires us to build the nodes and edges of the undirected graphical model and then determine their node and edge potentials.

- **Determining Nodes and Node Potentials**
  - The input sentence is passed into the Stanford POS Tagger to get the POS-tags of each word in the sentence.
  - For each content word (nouns, verbs, adjectives and adverbs) in the sentence, a node is created in the Markov Random Field, denoting the sense of that word. All possible senses of the corresponding word are the different values which the node can take.
  - Node Potentials denote the probability distribution of sense of the corresponding word. They are determined using the frequency of each sense of each of the above words in the Princeton WordNet.
- **Determining Edges and Edge Potentials**
  - The input sentence is passed into the Link Parser or Stanford Parser to determine the sense dependencies.
  - The dependencies obtained between the content words form the edges of the Markov Random Field.
  - Edge Potentials denote the probability of co-occurrence of particular senses of two dependent words. They can be determined using a variety of relatedness measures given in WordNet::Similarity (Pedersen, Patwardhan, and Michelizzi 2004) which include HirstStOnge (Hirst and St-Onge 1998), LeacockChodorow (Leacock and Chodorow 1998), Lesk (Banerjee and Pedersen 2002), WuPalmer (Wu and Palmer 1994), Resnik (Resnik 1995), Lin (Lin 1998), Jiang-Conrath (Jiang and Conrath 1997) and Path. All these relatedness measures use the Princeton Wordnet as the knowledge resource. They are appropriate for edge potentials as the relatedness between two synsets indicates their chances of co-occurrence.

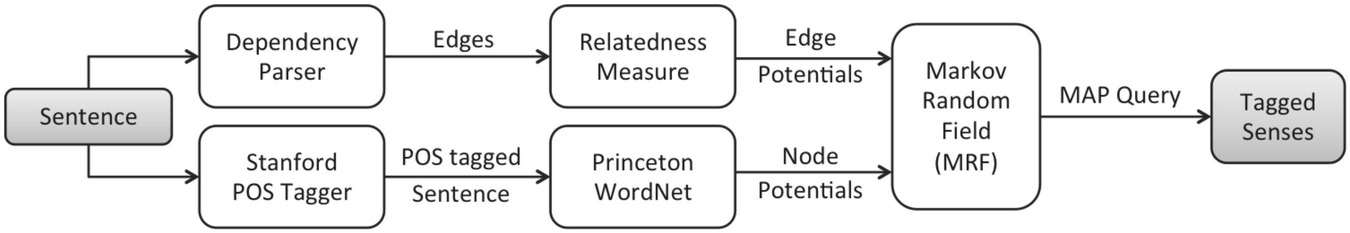

Figure 1: Block diagram of proposed algorithm

Maximizing the joint probability using MAP Inference Query

The nouns, verbs, adjectives and adverbs in the input sentence are the set of words to be disambiguated. Let these words be $W = \{w_1, w_2, ...w_n\}$ and their senses be $X = \{x_1, x_2, ...x_n\}$, respectively. Sense of each word x_i can take k_i possible values from the set $Y_i = \{y_i^1, y_i^2...y_i^{k_i}\}$, which are all the senses of the word w_i given its POS tag (determined using Stanford POS Tagger). Here, the Princeton WordNet is used as the sense inventory for each word. For example, consider the sentence,

"Cricket is a type of insect."

The words to be disambiguated are 'cricket', 'is', 'type' and 'insect'. So, $w_1 = Cricket, w_2 = is, w_3 = type, w_4 = insect$. x_1 is the sense of the word cricket, it can take two possible values $Y_1 = \{y_1^1, y_1^2\}$, given that it is a noun.

y_1^1 = "leaping insect; male makes chirping noises by rubbing the forewings together"

y_1^2 = "a game played with a ball and bat by two teams of 11 players; teams take turns trying to score runs"

We define $\Psi(x_i)$ as the node potential which represents the probability distribution of word x_i over its senses. This is calculated from the frequency of occurrence of each sense obtained from WordNet.

$$\Psi(x_i = y_i^a) \propto frequency(y_i^a) + 1 \; \forall a$$

Now, using Link Parser or Stanford Parser, we get the set of edges E between elements in X. We define $\Psi(x_i, x_j)$ as the joint probability distribution between senses of words x_i and x_j over the states they can be assigned. Let M be the relatedness measure used to find relatedness between two senses.

$$\Psi(x_i = y_i^a, x_j = y_j^b) \propto M(y_i^a, y_j^b) \; \forall a, b$$

These node and edge potentials are normalized to set the total probability to 1. Now, the joint probability distribution of senses of all words in X over each of its senses Y in a sentence is given by

$$\Psi(X) = \Psi(x_1, x_2...x_n)$$
$$= \prod_{x_i \in X} \Psi(x_i) \prod_{(x_i, x_j) \in E} \Psi(x_i, x_j)$$

The problem thus gets transformed into

$$\arg\max_{Y} \Psi(X = Y)$$

This is the MAP Inference query, which can be exactly solved in time exponential in tree width of the MRF. Note that the graph built using our method can only be as big as the number of words to be disambiguated in the sentence, but some other methods discussed in Section 2 build a much larger graph such as a sub-graph of the WordNet itself spanning all the words in the sentence. Furthermore, in practice, the tree width of graph built using our method is much smaller than even number of words to be disambiguated in a sentence as dependency parser keeps only useful edges (i.e. the edges which help in disambiguation) in order to avoid sense drift. On an average, Link Parser reduces the number of edges by about 91% on SensEval-2 and SensEval-3 all-words datasets. Note that we have reduced the complexity of the maximization problem by using dependency parser to remove unwanted edges, and this maximization is exact and optimal unlike other approximate or sub-optimal methods discussed in Section 2. The effect of this edge reduction on the labelling speed of our system is quantified in Section 7.

5 Reasoning behind proposed algorithm

This section describes the thought process behind devising the proposed algorithm.

Why graphical models?

Since we need to maximize the joint probability of occurrence of senses of all the words in the sentence, construction of graphical model is appropriate.

Why undirected?

Directed graphical models are suited for applications where there is a unilateral cause-effect relationship and the causes might be independent of one another, which is not the case with WSD as senses of dependent words affect each other. Therefore, we use Undirected Graphical Models (which are generally referred to as Markov Random Fields when subjected to a few assumptions). Sense of each word is represented as a node in the graph and edges are introduced to represent dependencies given by dependency parser. Expressing the WSD problem in such a manner allows us to leverage the existing algorithms which solve for exact inference in time exponential to the treewidth. **Treewidth** is defined as the size of the largest clique in a chordal completion of a graph. A chordal graph (or triangulated graph) is one which has no cycles of length greater than 4.

Why dependency parser?

Sense of word depends only on sense of few words in the sentence, and not all. We need to determine the words whose senses are dependent on each other. We argue that majority of these sense dependencies are captured in the syntactic structure of the sentence. Dependency parsers identify the dependencies between different words in the sentence using this syntactic structure. We have used two dependency parsers for our experiments:-

- **Link Parser** is a syntactic parser based on link grammar, an original theory of English syntax. It assigns a syntactic structure to the sentence which contains a set of labelled links or dependencies connecting pairs of words.
- **Stanford Parser** is also a syntactic parser which provides a list of typed dependencies between words in the given sentence. These dependencies provide a representation of grammatical relations between words in a sentence.

To understand the importance of dependency parser in identifying sense dependencies, consider the sentence,

"Bank is a financial institution."

Here, the word 'bank' is connected to the word 'institution' by the Link Parser. Hence, the word 'bank' is assigned the following sense

bank (a financial institution that accepts deposits and channels the money into lending activities)

Now, consider the sentence,

"There is a financial institution near the river bank."

Here, the word 'bank' is not connected to words 'institution' or 'financial' by the Link Parser. Hence, the word 'bank' is assigned the following sense

bank (sloping land (especially the slope beside a body of water))

If we do not use Link Parser in the above sentence, i.e. connect senses of all the words to each other, then the sense of word 'bank' drifts to the sense of "a financial institution". Therefore, we use the dependencies between words, obtained using the dependency parser, as the edges in our undirected graphical model in order to avoid sense drift.

6 Experiments and Results

We have tested our system on the SensEval-2 (Palmer et al. 2001), SensEval-3 (Snyder and Palmer 2004) and SemEval-2007 (Pradhan et al. 2007) English all-words WSD datasets. For each sentence in the dataset, we run the proposed algorithm as described in Section 4. We have used the Matlab UGM (Schmidt 2007) package for running the MAP Inference Query as it provides an optimal implementation using junction trees and message passing algorithm. We have used Path relatedness measure for our experiments. Experiments on subset of datasets showed that the difference due to choice of relatedness measure is not significant. Our system always marks a sense to a word to be disambiguated. Therefore, Precision and Recall are essentially equal. In Table 1 we compare our overall F1 scores with different Systems as mentioned in Section 2. In Table 2 we report the F1 scores on different parts of speech. Similar results using both Link Parser and Stanford Parser, as shown in Table 1, indicate that our system is not sensitive to the choice of dependency parser.

System	S2AW	S3AW	S07AW
Mih05	54.2	52.2	
Sinha07	57.6	53.6	
Tsatsa10	58.8	57.4	
Nav10		52.9	43.1
Agirre14	59.7	57.9	41.7
MRF-LP	**60.5**	58.6	**50.6**
MRF-SP	60.0	**58.7**	49.0

Table 1: Comparison of F1 scores with state-of-the-art unsupervised WSD systems on English all-words datasets of SensEval-2 (S2AW), SensEval-3 (S3AW) and SemEval-2007 (S07AW). MRF-LP and MRF-SP correspond to the proposed system using Link Parser and Stanford Parser, respectively. Best result in each column in **bold**.

SensEval-2 all-words dataset					
	All	N	V	Adj.	Adv.
Mih05	54.2	57.5	36.5	56.7	70.9
Sinha07	56.4	65.6	32.3	61.4	60.2
Agirre14	59.7	**70.3**	40.3	59.8	72.9
MRF-LP	**60.5**	66.9	**42.7**	**63.2**	**82.9**
SensEval-3 all-words dataset					
	All	N	V	Adj.	Adv.
Mih05	54.2	57.5	36.5	56.7	**100**
Sinha07	52.4	60.5	40.6	54.1	-
Nav07	-	61.9	36.1	62.8	-
Agirre14	57.9	65.3	47.2	**63.6**	96.3
MRF-LP	**58.6**	**65.8**	**50.1**	59.9	87
SemEval-2007 all-words dataset					
	All	N	V	Adj.	Adv.
Agirre14	41.7	56.0	35.3	-	-
MRF-LP	**50.6**	**63.8**	**43.4**	-	-

Table 2: Results on different POS tags of English datasets (F1). Best result in each column in **bold**.

We would like to highlight some difficulties faced while calculating the exact accuracies on the datasets used for comparison. Firstly, we have used WordNet 3.0 for our experiments as it was the latest when we began our experiments, but the original SensEval-2 and SensEval-3 all-words WSD Tasks were based on WordNet 1.7 and SemEval-2007 was based on WordNet 2.1. Some of the words which need to be tagged have been removed from WordNet 3.0. This includes words like anything, something, would, might, etc. which need not be disambiguated as they are not content words. Since these words were monosemous in WordNet 1.7, we have counted them as correct in our experiments.

Secondly, some words have been given ”U” tag in the gold standard of datasets. The ”U” tag stands for unassignable, which means the sense of the word is not included in the available senses, or perhaps that the annotators could not agree. Since, there are 89 such tags in SensEval-2, 34 in SensEval-3 and 10 in SemEval-2007, they have a considerable impact on the overall accuracy. Due to improvements in WordNet, many of these senses should not be unassignable using WordNet 3.0 as sense inventory. We report our accuracies separately, counting the ”U” tags as incorrect, ignoring them, and counting them as correct, in Table 3. Since previous works do not mention how they have handled the ”U” tag, we compare our accuracy ignoring the ”U” tag in Tables 1 and 2.

	”U” tag counted as		
	Incorrect	Ignored	Correct
MRF-LP			
S2AW	58.6	60.5	61.9
S3AW	57.6	58.6	59.3
S07AW	49.4	50.6	51.7
MRF-SP			
S2AW	58.1	60.0	61.4
S3AW	57.7	58.7	59.4
S07AW	47.9	49.0	50.1

Table 3: Overall F1 scores on counting ”U” tags as incorrect, ignored or correct.

7 Edge Reduction and Labelling Speed

As described earlier, using dependency parser, we determine the sense dependencies, thus removing unwanted edges and consequently reducing the treewidth of the MRF. Since our algorithm is exponential to the treewidth, edge reduction improves the speed of the system dramatically. Table 4 shows the percentage of edges removed by Link Parser and Stanford Parser and the time taken to run MAP Inference Query on all the sentences of the corresponding datasets. Test 1 and Test 2 are experimental systems which randomly reduce 70% and 60% of the edges, respectively. These experimental systems are used to quantize the effect of reduction of edges on the time taken to run the MAP Inference Query. If the edge reduction is less than 50%, system is unable to run on long sentences due to memory constraints. Thus, in order to run the exact MAP Inference Query, significant edge reduction is essential.

Due to this edge reduction, the proposed WSD system is significantly faster than the state-of-the-art. It labels the whole SensEval-2 dataset (2473 instances) in 55 seconds using Stanford Parser, which makes its labelling speed about 2700 instances per minute, while the state-of-the-art system (Agirre14) labels 70 instances per minute on SensEval-2. Table 5 provides a comparison of labelling speeds of the proposed system with the state-of-the-art. The labelling speed of the our system is averaged over 100 simulations on SensEval-2 dataset. Furthermore, SensEval-2 dataset does not require all the content words to be labelled, but our system labels all such words in the dataset, therefore, its actual labelling speed is even greater than that mentioned in Table 5. Stanford Parser gives significantly higher labelling speed as compared to Link Parser potentially because we have used the Link Parser ported to Java (JLinkGrammar), rather than the original software developed in C.

	SensEval-2		SensEval-3	
System	% Edges Reduced	Time Taken	% Edges Reduced	Time Taken
MRF-LP	91.47	6.39s	90.39	5.25s
MRF-SP	87.45	15.2s	84.57	18.2s
Test 1	70.00	70m	70.00	58m
Test 2	60.00	8h31m	60.00	6h46m

Table 4: Effect of edge reduction on time taken to run the MAP Query on the dataset. Test 1 and Test 2 are two experimental systems which randomly reduce 70% and 60% of the edges.

System	CPU	Mem	Labelling Speed (instances/min)
Agirre14-Best	2.66Ghz	16GB	70
Agirre14-Fastest	2.66Ghz	16GB	684
MRF-LP	2.4Ghz	8GB	1144
MRF-SP	2.4Ghz	8GB	2698

Table 5: Comparison of labelling speed with the state-of-the-art. Agirre14-Best refers to PPR_{w2w} system which gives the best accuracy and Agirre14-Fastest refers to full graph PPR system which was the fastest.

8 Web Interface

We have developed a web interface of our Unsupervised WSD Algorithm for visualizing the sense dependencies captured by Link Parser or Stanford Parser, error analysis and parameter optimization. The input and output of the web interface have been discussed in this section. Figure 2 shows a screen shot of the web interface.

Input

Input options to the web interface include the following:-

1. Query sentence: The sentence to be disambiguated.
2. Dependency Parser: Link Parser or Stanford Parser can be selected as the dependency parser.
3. Relatedness Measure: The different relatedness measures available are HirstStOnge, LeacockChodorow, Lesk, Wu-Palmer, Resnik, JiangConrath, Lin and Path.

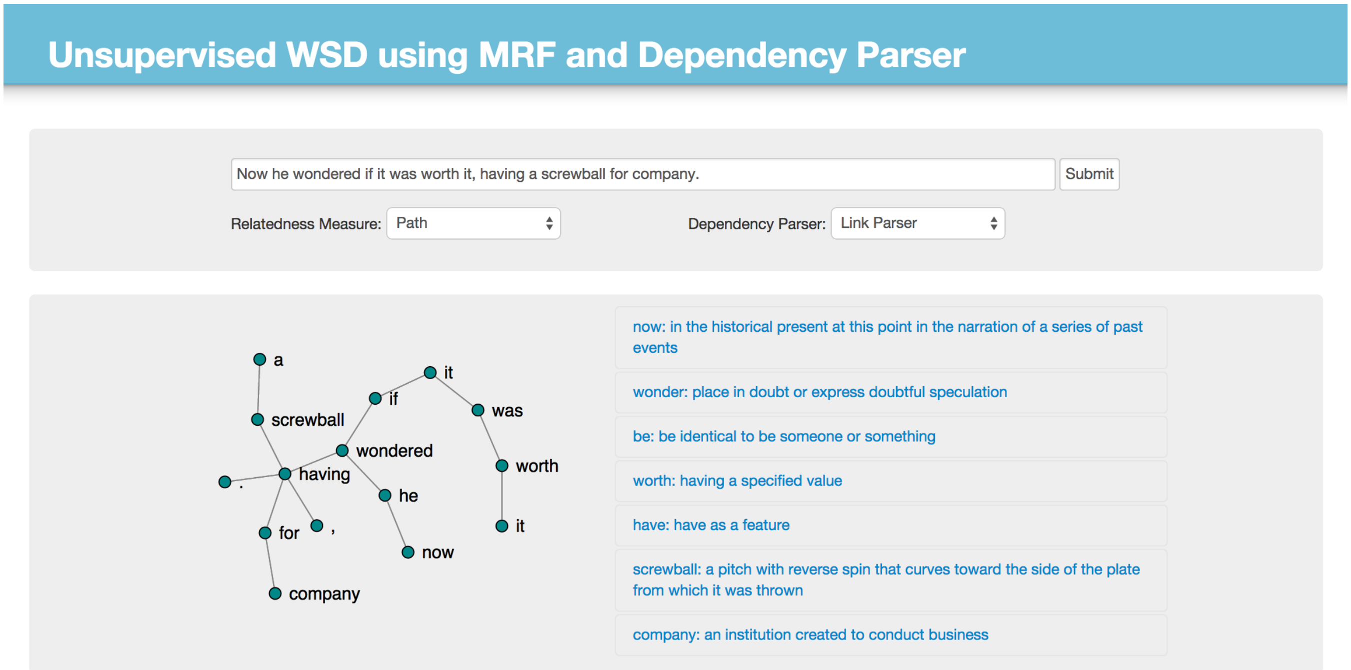

Figure 2: Screenshot of the web interface

Output

The output to the web interface includes the following:-

1. Tagged Senses of the content words in the sentence.
2. Details of tagged senses can be seen by clicking on the sense. It includes the POS tag of the word, Sense ID as given in Princeton WordNet 3.0 and WordNet hyperlink to possible senses of the word.
3. Dependency Graph: Graph of sense dependencies determined using Link Parser or Stanford Parser.

9 Conclusion and Future Work

We propose a graph-based unsupervised WSD system which maximizes the total joint probability of all the senses in the context. Our algorithm removes unwanted edges in the graph using Link Parser or Stanford Parser, thus reducing the exponential time complexity of the system. Removal of undesirable edges not only limits sense drift, leading to significant improvement in accuracy, but also has dramatic effects on the labelling speed of the proposed system. It beats the state-of-the-art on SensEval-2, SensEval-3 and SemEval-2007 all-words datasets, while being more than 35 times faster. Link Parser and Stanford Parser give similar results, showing dependency relations are useful for Word Sense Disambiguation. We have also developed a web interface for experiments, visualization, error analysis, optimization and further improvements.

In future, we intend to devise intelligent techniques to improve the edges in the graph, as sense dependencies are not always captured in syntactic structure of the sentence, and thus not identified by dependency parser. For example, in the sentence "The man withdrew money from the bank", money and bank not related in syntactic structure. Incorporating other techniques to find sense dependencies could lead to further improvement of the algorithm.

References

Agirre, E., and Rigau, G. 1996. Word sense disambiguation using conceptual density. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, 16–22. Association for Computational Linguistics.

Agirre, E.; López de Lacalle, O.; and Soroa, A. 2014. Random walks for knowledge-based word sense disambiguation. *Comput. Linguist.* 40(1):57–84.

Banerjee, S., and Pedersen, T. 2002. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, 136–145. London, UK, UK: Springer-Verlag.

Chan, Y. S.; Ng, H. T.; and Chiang, D. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 33–40. Prague, Czech Republic: Association for Computational Linguistics.

Cowie, J.; Guthrie, J.; and Guthrie, L. 1992. Lexical disambiguation using simulated annealing. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 1*, 359–365. Stroudsburg, PA, USA: Association for Computational Linguistics.

Hirst, G., and St-Onge, D. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press. 305–332.

Jiang, J. J., and Conrath, D. W. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR* cmp-lg/9709008.

Jordan, M. I., and Weiss, Y. 2002. Probabilistic inference in graphical models. In *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 2nd edition.

Leacock, C., and Chodorow, M. 1998. Combining local context and wordnet similarity for word sense identification. In *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press. 265–283.

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

Marneffe, M.; Maccartney, B.; and Manning, C. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*. Genoa, Italy: European Language Resources Association. ACL Anthology Identifier: L06-1260.

Mihalcea, R. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 411–418. Association for Computational Linguistics.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Communications of the ACM* 38(11):39–41.

Navigli, R., and Lapata, M. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, 1683–1688. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Navigli, R., and Lapata, M. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(4):678–692.

Palmer, M.; Fellbaum, C.; Cotton, S.; Delfs, L.; and Dang, H. T. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, 21–24. Toulouse, France: Association for Computational Linguistics.

Panagiotopoulou, V.; Varlamis, I.; Androutsopoulos, I.; and Tsatsaronis, G. 2012. Word sense disambiguation as an integer linear programming problem. In *Proceedings of the 7th Hellenic Conference on Artificial Intelligence: Theories and Applications*, 33–40. Berlin, Heidelberg: Springer-Verlag.

Patwardhan, S.; Banerjee, S.; and Pedersen, T. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing*, 241–257. Berlin, Heidelberg: Springer-Verlag.

Pedersen, T.; Patwardhan, S.; and Michelizzi, J. 2004. Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, 38–41. Stroudsburg, PA, USA: Association for Computational Linguistics.

Pradhan, S. S.; Loper, E.; Dligach, D.; and Palmer, M. 2007. Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, 87–92. Stroudsburg, PA, USA: Association for Computational Linguistics.

Ramakrishnan, G.; Jadhav, A.; Joshi, A.; Chakrabarti, S.; and Bhattacharyya, P. 2003. Question answering via bayesian inference on lexical relations. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, 1–10. Stroudsburg, PA, USA: Association for Computational Linguistics.

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, 448–453. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Schmidt, M. 2007. UGM: Matlab code for undirected graphical models. http://www.cs.ubc.ca/~schmidtm/Software/UGM.html.

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

Sleator, D., and Temperley, D. 1993. Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*.

Snyder, B., and Palmer, M. 2004. The english all-words task. In Mihalcea, R., and Edmonds, P., eds., *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 41–43. Barcelona, Spain: Association for Computational Linguistics.

Toutanova, K.; Klein, D.; Manning, C. D.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, 173–180. Stroudsburg, PA, USA: Association for Computational Linguistics.

Wu, Z., and Palmer, M. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, 133–138. Stroudsburg, PA, USA: Association for Computational Linguistics.

Zhong, Z., and Ng, H. T. 2012. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, 273–282. Stroudsburg, PA, USA: Association for Computational Linguistics.