Word Sense Disambiguation Using Numerical Constraint Satisfaction

Julia Komissarchik*
Claire Cardie

Department of Computer Science
Cornell University
Ithaca, NY 14853.

Abstract

In this paper we present a description and evaluation of a word sense disambiguation mechanism that is an extension of the work by Ide and Veronis [3]. The mechanism's underlying approach is a numerical constraint satisfaction algorithm applied to the network built using an on-line dictionary. In this work we aim to duplicate the original results and extend the method to large-scale applications. To determine the scalability of the approach, we evaluate the mechanism on 20 ambiguous words taken in their contexts of different size from the Brown corpus. We offer a discussion of encountered difficulties and describe several possible directions that can be investigated to improve the mechanism's performance and scalability.

1 Introduction

To understand a text, a natural language processing (NLP) system must determine the meaning of the words that comprise the text — both a difficult and important task. More specifically, the task of word sense disambiguation (WSD) is to consider a polysemous word (a word that has more than one meaning) in context and determine which of the word's meanings is the most appropriate for the given context.

Among other factors, the granularity of the sense distinctions determines the difficulty of WSD. Consider for example, the following sentence taken from the Brown corpus: "I can't tell you now the distance in lightyears or what star this was, though you can see it with the naked eye at night in this hemisphere".

*Supported by the NSF Grant GER-9454149. Email: `juliak@cs.cornell.edu`

Here the word "star" is used in its "celestial object" sense. The Collins English Dictionary, however, distinguishes the sense of "star" as an object in the sky from the sense of "star" as an astronomical object; and in this case it is difficult even for a human observer to decide which of the senses is more appropriate.

There are several approaches to automatic WSD. They include methods that use hand-crafted, detailed "word experts" [6], relaxation network based methods [1, 7], methods that use machine-readable dictionaries [4], case-based reasoning methods [5], and statistical methods [8, 9]

The approach taken here is based on work by Ide and Veronis [3], that views WSD as a constraint satisfaction problem: the meaning of a word is constrained by the meanings of its neighbors. In particular, they employ a localist connectionist network that is automatically constructed using a machine readable dictionary. This method has several distinctive features: (a) it is an unsupervised method that requires no training corpus; (b) it disambiguates all the words in the text simultaneously rather than disambiguating them one by one; (c) it does not require a complicated statistical model; (d) although it is a knowledge-based method, it requires no hand-coded knowledge base or disambiguation rules.

To evaluate their method, Ide and Veronis implemented a prototype of the WSD mechanism and tested it on 138 inputs of two words each. They obtained 85% accuracy. In this paper we extend the approach and evaluate it in a more practical setting, using a real corpus and input texts of a significantly larger size.

Section 2 describes our implementation of the mechanism; Section 3 presents the evaluation of the system on different inputs; Section 4 discusses the results.

2 A Connectionist Approach to Word Sense Disambiguation

Our implementation of the Ide and Veronis word sense disambiguation mechanism consists of two independent phases: construction and disambiguation. In the network construction phase, the program takes the Collins English Dictionary as input, preprocesses it, and outputs a relaxation network that corresponds to the dictionary. In the disambiguation phase, the program reads in a preprocessed text, initializes the nodes corresponding to the words in the sentence, applies a spreading activation procedure, and outputs the text with sense tags attached to the words.

Section 2.1 describes the general structure of relaxation networks and the spreading activation procedure; Section 2.2 and Section 2.3 describe the particular structure of the relaxation network used in our WSD mechanism.

star	1.	
	pos	n.
	def	any of a vast number of celestial objects that are visible in the clear night sky as points of light.
...		
star	4.	
	pos	n.
	def	an emblem shaped like a conventionalized star, usually with five or more points, often used as a symbol of rank, an award, etc.
...		

Table 1: A Portion of the Collins English Dictionary Entry for the Word "star", (pos = part of speech; def = definition)

2.1 Relaxation Networks and Spreading Activation Algorithms

A relaxation network is a localist connectionist network that consists of nodes and links between them [2]. Each node has an activation level — a "potential" that can take on a value between 0 and 10. Each link has a weight associated with it. Links are of two types — positively weighted (excitatory) links or negatively weighted (inhibitory) links. Excitatory links propagate the energy in the system, whereas inhibitory links provide competition between nodes. During the activation of the network, a subset of the nodes receives a non-zero potential and energy is propagated through the network. The potential of a node, a_j, at time $t+1$ is

$$a_j(t)+\text{Exc - Inh}$$

where Exc is the weighted sum of the excitatory inputs and Inh is the weighted sum of the inhibitory inputs. Details of the spreading activation algorithm will be described in section 2.3.

2.2 Network Architecture

Our network is based on the on-line Collins English Dictionary (CED). In the CED each entry consists of a word token and a list of possible meanings this word token can have; each of the meanings is defined with one or more sentences. A portion of the CED entry for the word "star" is given in Table 1. Figure 1 shows a subset of the network that corresponds to this portion of the dictionary. The network has two types of nodes: for each entry in the dictionary there is a *word node* that is labeled with the entry's word token. For each word meaning defined in an entry there is a corresponding *sense node*. Each word node is connected via excitatory links to the one or more sense nodes that correspond

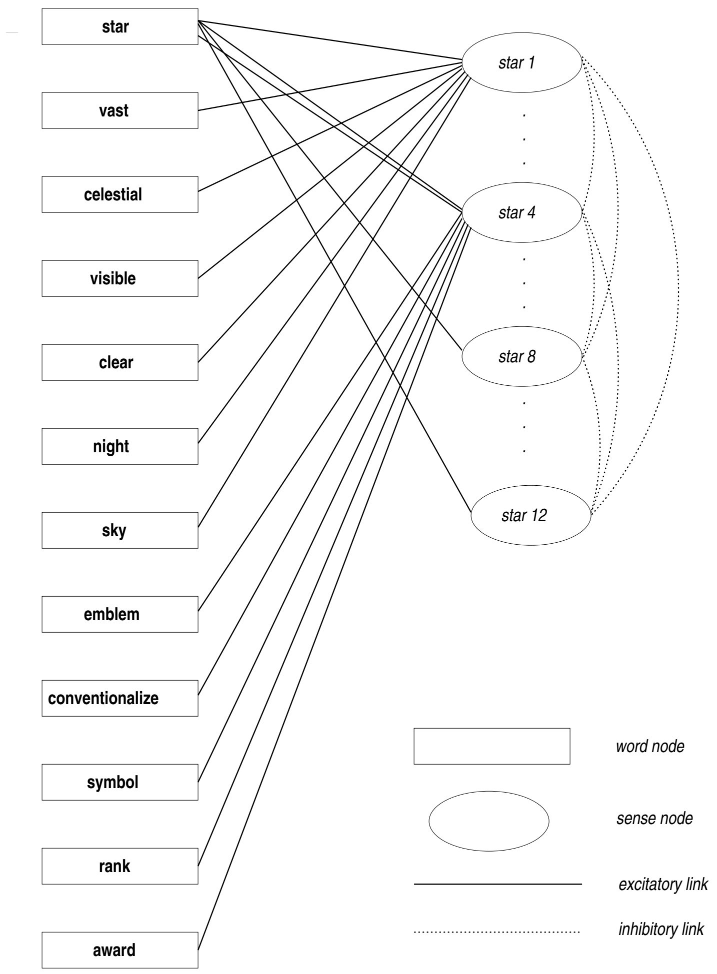

Figure 1: A Portion of the Network Corresponding to Table 1.

star	1	vast celestial visible clear night sky
...		
star	4	emblem conventionalize star symbol rank award
...		

Table 2: Preprocessed Portion of a Dictionary Entry for “star”

to the meanings defined in the entry. The sense nodes are also connected via excitatory links to its “defining” word nodes — nodes whose corresponding word tokens appear in the definition of the word’s senses. Finally, each of the sense nodes is connected via inhibitory links to the sense nodes that correspond to other meanings defined in the same entry. Inhibitory links facilitate the competition between senses of the same word.

The resulting network that represents the entire CED has over 130,000 word nodes and over 160,000 sense nodes.

While constructing the network, we would like to extract as much useful information from the dictionary as possible while skipping over the words that are likely to contribute little to the WSD process. We would also like to avoid discriminating between different morphological forms of the same word. Therefore, a preprocessing stage precedes network construction:

- Rudimentary morphological normalization of the entries’ definitions is carried out by removing common endings and suffixes (*ing, ed, s, ’s, es, en*).
- *Stop words* are removed. For our purposes, stop words are words that occur in 1000 or more dictionary entries.
- *Stop phrases* are removed. For our purposes, stop phrases are phrases that appear often in the definitions, and are used for classification purposes; e.g. “a word for”, “Short for”, ...

An example of a preprocessed dictionary entry is given in Table 2.

Prior to the disambiguation process, the input text is subjected to the same preprocessing — the text is morphologically normalized and the stop words are removed.

2.3 Spreading Activation Algorithm

In this section we describe the spreading activation procedure that is used in the current implementation.

Let $a_i(t)$ be the potential of a node a_i at time t. Let $w_{i,j}$ be the weight of an edge $e_{i,j}$ from node a_i to node a_j. A node’s potential at time t has a real value between 0 and 10. Each excitatory link is assigned the same excitatory weight; similarly, each inhibitory link is assigned the same inhibitory weight. During the initialization of the network, the word nodes that correspond to tokens in the preprocessed input text are given a potential of 10.

The energy is propagated through the network until there is a winner among the competing senses for every word in the input text or until the number of iterations is above certain threshold and there is still no winner among the competing senses of the input text words. During each iteration t the potential of a_i is calculated as follows:

$$a_i(t+1) = a_i(t) * \mathrm{DecayRate} + n_i(t)$$

Where

- *DecayRate* is a real value between 0 and 1. It is introduced to ensure that if a node does not receive the input from its neighbors, it will eventually return to the zero potential level.
- $n_i(t)$ refers to the influence of the node's neighbors and is equal to:

  $\sum_{j:\exists e_{j,i}} w_{j,i} * a_j(t)/(\text{total } \# \text{ of outgoing edges of } a_j)$
- the energy propagated through an edge is normalized by the number of outgoing edges to guarantee that a node with a lot of outgoing edges will propagate the same total energy as a node with a few outgoing edges.
- if a node's potential falls below 0, the potential of 0 is assigned to that node at the current iteration. If a node's potential rises above 10, the potential of 10 is assigned to that node at the current iteration.
- the winner sense node is defined as a node with the highest potential among the competing senses.

The implemented network has several parameters:

- the minimum and the maximum level of potential that a node can have;
- the excitatory edge weight;
- the inhibitory edge weight;
- the decay rate;
- the "liveness" threshold — a value above which a node's potential must stay in order to remain active;
- maximum number of iterations of the spreading activation algorithm.

Analysis of the structure of the net and monitoring of the spreading activation procedure has been performed using visualization tools created specifically for these purposes.

2.4 Differences with Ide and Veronis

The current implementation of the network (described above) differs from the original Ide and Veronis implementation in several aspects.

- In the current implementation, the constructed network corresponds to the complete CED rather than a subset of the dictionary that depends on the input text to be disambiguated. On average, such a network is two orders of magnitude smaller than our network.
- The current implementation takes input texts of any size rather than an input of two words (i.e., one ambiguous word plus one context word).
- Ide and Veronis state that the outgoing energy of each active node is normalized by a simple decreasing function of the number of outgoing connections from a given node. We implement this energy as equal to 1/(total # of outgoing edges). This is done to ensure that the total energy passed from a node is equivalent to that node's potential.
- Ide and Veronis introduce a bias towards the first sense within a group of homographs. In our implementation, all senses are treated equally.

3 Experiments

We performed two sets of experiments. The first set of experiments was designed to duplicate the tests presented by Ide and Veronis as much as possible. In the second set of experiments, the complete network was used for WSD of words in the texts of various length; this set was designed to evaluate the applicability of the approach to large-scale WSD problems. The sections below describe each set in turn.

3.1 Small Network

In these experiments the spreading activation algorithm is applied to a subnetwork rather than the entire CED network. That subnetwork is essentially 2.5 levels of words and their senses. It consists of the nodes that correspond to: (1) words in the input text, (2) all of their CED senses, (3) words that make up the definitions of these senses, (4) the senses of these defining words, and (5) words that make up definitions of the latter senses. The size of this subnetwork for the input *ash residue*, for example, is 1354 nodes rather than 290,000 nodes of the full CED network. We test a number of network parameter settings: [1]

[1] Ide and Veronis do not provide their network parameters

	ambiguous word	context word
1	pen	goat
2	pen	page
3	pen	book
4	ash	residue
5	ash	fire
6	ash	tobacco
7	ash	tree
8	ash	branch
9	ash	lawn
10	bay	peninsula
11	bay	cape
12	bay	sail
13	bay	wall
14	bay	window
15	bay	house

Table 3: Word pairs listed by Ide and Veronis

Parameter	Optimal Value	Tested over Range
Range of Node's Potential	0..10	0..1 - 0..10
Excitatory Links Weight	.5	.1 - 2
Inhibitory Links Weight	2	.5 - 2
Decay Rate	.75	.6 - .9
"Liveness" Threshold	0	0 - .1
Maximum # of Iterations	1000	20 - 1000

Ide and Veronis test their prototype on 138 examples. Each example consists of a pair of words — an ambiguous word and a context word. The context words are hand-picked from the Brown corpus. Their results are summarized in Table 4, where efficiency is the percentage of the words disambiguated correctly and reliability is the percentage of the correct answers out of all of the answers for which a conclusion was reached.

Ide and Veronis explicitly list 15 of the 138 inputs, these are shown in Table 3. The results of our experiments on pairs from Table 3 are given in Table 5. For the three pairs that were described in detail in [3] in our experiments two out of three were disambiguated correctly. On average it took the network 16 iterations to converge.

[2] The results presented in [3] and the results presented here are given with respect to the groups of homographs. For example, during the disambiguation of ash all meanings of the word ash, as defined in CED, are considered. However, when evaluating the disambiguation results, the the meanings of the word *ash* as a tree and *ash* as residue are discriminated, but the senses of *ash* as a tree and *ash* as wood of an ash tree are treated as the same.

Number of Examples	Correct[2]	Incorrect	Unanswered	Efficiency (Recall)	Reliability (Precision)
138	118	4	16	$\frac{118}{138} = 85\%$	$\frac{118}{122} = 97\%$

Table 4: Prototype Performance (Ide and Veronis)

Number of Examples	Correct[2]	Incorrect	Unanswered	Efficiency (Recall)	Reliability (Precision)
15	6	3	6	$\frac{6}{15} = 40\%$	$\frac{6}{9} = 67\%$

Table 5: Our Systems' Performance on the Words from Table 3

3.2 Large Network

In the second set of experiments a network corresponding to the full CED is considered. The network is used for the disambiguation of 20 commonly tested polysemous words from the WSD literature. All instances of the test words are extracted from the Brown corpus along with windows of surrounding text. The Brown corpus is chosen for testing because it includes a diverse set of material and is a representative sample of actual language use.

We compare our methods performance with the results reported by Yarowsky [8, 9]. The statistical method developed by Yarowsky achieves one of the best performance results in WSD. Yarowsky has tested his method on the same set of ambiguous words extracting instances of these words along with windows of surrounding text from Grolier's Encyclopedia.

The parameters for the experiments are:

Parameter	Optimal Value	Tested over Range
Range of Node's Potential	0..10	0..1 - 0..10
Excitatory Links Weight	.5	.1 - 2
Inhibitory Links Weight	2	.5 - 2
Decay Rate	.75	.6 - .9
"Liveness" Threshold	0	0 - .1
Maximum # of Iterations	1000	20 - 1000

The experiments have been performed for three different sizes of the win-

word	total # of instances	correct	% accuracy	Yarowsky '92 & '95 (examples from Grolier's)
bass	16	8	50%	99%
bow	12	2	17%	91%
cone	6	2	34%	77%
drug	19	13	68%	93%
duty	47	23	49%	96%
galley	4	0	0%	95%
interest	309	168	54%	72%
issue	140	12	9%	94%
mole	2	0	0%	99%
motion	54	11	20%	98%
palm	18	4	22%	97%
plant	119	24	20%	98%
poach	1	1	100%	97%
sake	41	36	88%	96%
sentence	32	5	15%	98%
slug	7	0	0%	97%
space	158	82	52%	94%
star	16	4	25%	96%
tank	11	4	36%	97%
taste	55	30	55%	93%
total	**1154**	**429**	**37%**	**94%**

Table 6: Results of the Experiments

dows of the surrounding text: 100, 200 and 300 words. On average, a window of 200 words gives the best performance. For the testing purposes, the spreading activation over the network is performed until there is a winner among competing senses of the chosen ambiguous word **only**. The results of disambiguation with the optimal parameters are given in Table 6.

A direct comparison between our approach and Yarowsky's approach is difficult since it is essential to take into account that the implemented mechanisms consider different levels of sense granularity, are conceptually different and are tested on examples from different corpora:

- Examples in our experiments came from the Brown corpus, whereas Yarowsky uses examples from Grolier's Encyclopedia.

- Yarowsky reduces the problem of word sense distinction to the problem of category discrimination.

Source	Average Number of Senses
Collins English Dictionary	10.85
Roget's Thesaurus Categories	2.8

Table 7: Average Number of Senses

- Yarowsky uses Roget's Thesaurus categories for sense distinction; these categories are significantly coarser than CED categories. For example, for the word *plant*, Yarowsky lists two senses: living and factory. Whereas CED defines 13 different senses of the word *plant*. The average number of senses for the 20 listed words is given in Table 7.

To partially compensate for the discrepancy of the sense granularity, the output tags are mapped to the senses as reported by Yarowsky [8, 9]. To make the results more comparable it is necessary to use the more coarse Roget's Thesaurus sense distinction when constructing the network. The performance will likely improve if the network is built with the coarser senses.

4 Discussion

The goal of the first set of experiments was to repeat the experiments presented by Ide and Veronis [3]. The experiments demonstrate somewhat similar performance of the current implementation with the one described by Ide and Veronis; however, it was disconcerting not to be able to duplicate the results. Without additional details regarding the original implementation (especially the network parameter settings), obtaining identical results is difficult.

The goal of the second set of experiments was to determine the scalability of the method to large applications. The results of the experiments highlight several restrictions on the applicability of the implemented method to large scale word-sense disambiguation problems.

- As in the experiments with the smaller network, selecting parameters for optimal performance is difficult.
- The fine-grained distinctions between senses as defined in CED make the task difficult.
- Spreading activation procedure is computationally expensive (from 1 to over 30 min per input)

There are several modifications that might improve the performance of the implemented approach:

- Build the network using coarser sense distinctions by combining related senses from the Collins English Dictionary into the few main senses as defined in the Roget's Thesaurus.

- Reduce the number of possible senses for each word in the text, by applying a part of speech tagger to the input text before applying the WSD algorithm and splitting each word node (together with the connections to its sense nodes), into several word nodes that incorporate part of speech information.

- Reduce the number of the "defining" words of a sense by parsing the dictionary definitions. For example, rather than using all of the words in a definition, the mechanism will consider noun phrases only.

5 Conclusion

In this paper we presented a mechanism for word sense disambiguation. This mechanism is an extension of the approach introduced by Ide and Veronis [3]. The prototype that has been implemented by Ide and Veronis has given the WSD accuracy of 85% on 138 pairs of words. As expected, when developed into a large scale system and applied to WSD in the real text of various length the accuracy drops. In particular, the mechanism has the WSD accuracy of 37% on all instances of the selected 20 polysemous words in the Brown corpus. In the paper, we offer a detailed comparison of the mechanism performance with the performance that is reported by Yarowsky[8, 9]. While our initial attempt to extend the mechanism to large-scale applications yielded disappointing results, we outline several directions that can be investigated to improve the performance of the mechanism in practice.

References

[1] Cotrell, G.W., Small, S.L. (1983) *A Connectionist Scheme for Modeling Word Sense Disambiguation*, Cognition and Brain Theory, 6(1), 89-120,

[2] Feldman, J.A. and Ballard, D.H. (1982) *Connectionist models and their properties*, Cognitive Science, 6, 205-254.

[3] Ide and Veronis, *Large Neural Networks for the Resolution of Lexical Ambiguity*, Computational Lexical Semantics, 251-269.

[4] Lesk, M. (1986) *Automated Sense Disambiguation Using Machine-readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone.* Proceedings of the 1986 SIGDOC Conference.

[5] Ng, H.T., Lee, H.B. (1996) *Integrating Multiple Knowledge Sources to Disambiguate Word Sense: an Exemplar-Based Approach*, 34th Annual Meeting of Association for Computational Linguistics, 40-47.

[6] Small, S. and Rieger, C (1982) *Parsing and Comprehending with Word Experts (a Theory and its Realization)* Strategies for Natural Language Processing, Lehnert, W. and Ringle, M. eds.

[7] Waltz, D.L., Pollack, J.B. (1985) *Massively Parallel Parsing: a Strongly Interactive Model of Natural Language Interpretation*, Cognitive Science, 9, 51-74.

[8] D. Yarowsky. (1992) *Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora*, Proceedings, COLING-92, 454-460.

[9] D. Yarowsky. (1995) *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods*, 33rd Annual Meeting of Association for Computational Linguistics, 189-196.