UOY: A Hypergraph Model For Word Sense Induction & Disambiguation

Ioannis P. Klapaftis
University of York
Department of Computer Science
giannis@cs.york.ac.uk

Suresh Manandhar
University of York
Department of Computer Science
suresh@cs.york.ac.uk

Abstract

This paper is an outcome of ongoing research and presents an unsupervised method for automatic word sense induction (WSI) and disambiguation (WSD). The induction algorithm is based on modeling the co-occurrences of two or more words using hypergraphs. WSI takes place by detecting high-density components in the co-occurrence hypergraphs. WSD assigns to each induced cluster a score equal to the sum of weights of its hyperedges found in the local context of the target word. Our system participates in SemEval-2007 word sense induction and discrimination task.

1 Introduction

The majority of both supervised and unsupervised approaches to WSD is based on the "fixed-list" of senses paradigm where the senses of a target word is a closed list of definitions coming from a standard dictionary (Agirre et al., 2006). Lexicographers have long warned about the problems of such an approach, since dictionaries are not suited to this task; they often contain general definitions, they suffer from the lack of explicit semantic and topical relations or interconnections, and they often do not reflect the exact content of the context, in which the target word appears (Veronis, 2004).

To overcome this limitation, unsupervised WSD has moved towards inducing the senses of a target word directly from a corpus, and then disambiguating each instance of it. Most of the work in WSI is based on the vector space model, where the context of each instance of a target word is represented as a vector of features (e.g second-order word co-occurrences) (Schutze, 1998; Purandare and Pedersen, 2004). These vectors are clustered and the resulting clusters represent the induced senses. However, as shown experimentally in (Veronis, 2004), vector-based techniques are unable to detect low-frequency senses of a target word.

Recently, graph-based methods were employed in WSI to isolate highly infrequent senses of a target word. HyperLex (Veronis, 2004) and the adaptation of PageRank (Brin and Page, 1998) in (Agirre et al., 2006) have been shown to outperform the most frequent sense (MFS) baseline in terms of supervised recall, but they still fall short of supervised WSD systems.

Graph-based approaches operate on a 2-dimensional space, assuming a one-to-one relationship between co-occurring words. However, this assumption is insufficient, taking into account the fact that two or more words are usually combined to form a relationship of concepts in the context. Additionally, graph-based approaches fail to model and exploit the existence of collocations or terms consisting of more than two words.

This paper proposes a method for WSI, which is based on a hypergraph model operating on a n-dimensional space. In such a model, co-occurrences of two or more words are represented using weighted hyperedges. A hyperedge is a more expressive representation than a simple edge, because it is able to capture the information shared by two or more words. Our system participates in

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 414–417,
Prague, June 2007. ©2007 Association for Computational Linguistics

SemEval-2007 word sense induction and discrimination task (SWSID) (Agirre and Soroa, 2007).

2 Sense Induction & Disambiguation

This section presents the induction and disambiguation algorithms.

2.1 Sense Induction

2.1.1 The Hypergraph Model

A hypergraph $H = (V, F)$ is a generalization of a graph, which consists of a set of vertices V and a set of hyperedges F; each hyperedge is a subset of vertices. While an edge relates 2 vertices, a hyperedge relates n vertices (where $n \geq 1$). In our problem, we represent each word by a vertex and any set of co-occurring related words by a hyperedge. In our approach, we restrict hyperedges to 2, 3 or 4 words. Figure 1 shows an example of an abstract hypergraph model [1].

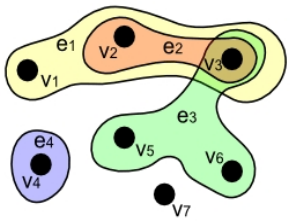

Figure 1: An example of a Hypergraph

The *degree* of a vertex is the number of hyperedges it belongs to, and the degree of a hyperedge is the number of vertices it contains. A path in the hypergraph model is a sequence of vertices and hyperedges such as $v_1, f_1, ..., v_{i-1}, f_{i-1}, v_i$, where v_k are vertices, f_k are hyperedges, each hyperedge f_k contains vertices to its left and right in the path and no hyperedge or vertex is repeated. The length of a path is the number of hyperedges it contains, the distance between two vertices is the shortest path between them and the distance between two hyperedges is the minimum distance of all the pairs of their vertices.

2.1.2 Building The Hypergraph

Let bp be the base corpus from which we induce the senses of a target word tw. Our bp consists of BNC and all the SWSID paragraphs containing the target word. The total size of bp is 2000 paragraphs. Note that if SWSID paragraphs of tw are more than 2000, BNC is not used.

In order to build the hypergraph, tw is removed from bp and each paragraph p_i is POS-tagged. Following the example in (Agirre et al., 2006), only nouns are kept and lemmatised. We apply two filtering heuristics. The first one is the minimum frequency of nouns (parameter p_1), and the second one is the minimum size of a paragraph (parameter p_2).

A key problem at this stage is the determination of related vertices (nouns), which can be grouped into hyperedges and the weighting of each such hyperedge. We deal with this problem by using association rules (Agrawal and Srikant, 1994). Frequent hyperedges are detected by calculating *support*, which should exceed a user-defined threshold (parameter p_3).

Let f be a candidate hyperedge and a, b, c its vertices. Then $freq(a, b, c)$ is the number of paragraphs in bp, which contain all the vertices of f, and n is the total size of bp. *Support* of f is shown in Equation 1.

$$support(f) = \frac{freq(a, b, c)}{n} \quad (1)$$

The weight assigned to each collected hyperedge, f, is the average of m calculated *confidences*, where m is the size of f. Let f be a hyperedge containing the vertices a, b, c. The *confidence* for the rule $r_0 = \{a, b\} => \{c\}$ is defined in Equation 2.

$$confidence(r_0) = \frac{freq(a, b, c)}{freq(a, b)} \quad (2)$$

Since there is a three-way relationship among a, b and c, we have two more rules $r_1 = \{a, c\} => \{b\}$ and $r_2 = \{b, c\} => \{a\}$. Hence, the weighting of f is the average of the 3 calculated *confidences*. We apply a filtering heuristic (parameter p_4) to remove hyperedges with low weights from the hypergraph. At the end of this stage, the constructed hypergraph is reduced, so that our hypergraph model agrees with the one described in subsection 2.1.1.

2.1.3 Extracting Senses

Preliminary experiments on 10 nouns of SensEval-3 English lexical-sample task (Mihalcea et al., 2004) (S3LS), suggested that our hypergraphs

[1] Image was taken from Wikipedia (Rocchini, 2006)

are small-world networks, since they exhibited a high clustering coefficient and a small average path length. Furthermore, the frequency of vertices with a given degree plotted against the degree showed that our hypergraphs satisfy a power-law distribution $P(d) = c * d^{-\alpha}$, where d is the vertex degree, $P(d)$ is the frequency of vertices with degree d. Figure 2 shows the log-log plot for the noun *difference* of S3LS.

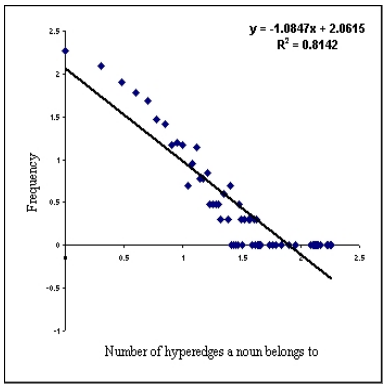

Figure 2: Log-log plot for the noun *difference*.

In order to extract the senses of the target word, we modify the HyperLex algorithm (Veronis, 2004) for selecting the root hubs of the hypergraph as follows. At each step, the algorithm finds the vertex v_i with the highest degree, which is selected as a root hub, according to two criteria.

The first one is the minimum number of hyperedges it belongs to (parameter p_5), and the second is the average weight of the first p_5 hyperedges (parameter p_6) [2]. If these criteria are satisfied, then hyperedges containing v_i are grouped to a single cluster c_j (new sense) with a 0 distance from v_i, and removed from the hypergraph. The process stops, when there is no vertex eligible to be a root hub.

Each remaining hyperedge, f_k, is assigned to the cluster, c_j, closest to it, by calculating the minimum distance between f_k and each hyperedge of c_j as defined in subsection 2.1.1. The weight assigned to f_k is inversely proportional to its distance from c_j.

2.2 Word Sense Disambiguation

Given an instance of the target word, tw, paragraph p_i containing tw is POS-tagged, nouns are kept and lemmatised. Next, each induced cluster c_j is assigned a score equal to the sum of weights of its hyperedges found in p_i.

[2]Hyperedges are sorted in decreasing order of weight

3 Evaluation

3.1 Preliminary Experiments

This method is an outcome of ongoing research. Due to time restrictions we were able to test and tune (Table 1), but not optimize, our system only on a very small set of nouns of S3LS targeting at a high supervised recall. Our supervised recall on the 10 first nouns of S3LS was 66.8%, 9.8% points above the MFS baseline.

Parameter	Value
p_1:Minimum frequency of a noun	8
p_2:Minimum size of a paragraph	4
p_3:Support threshold	0.002
p_4:Average confidence threshold	0.2
p_5:Minimum number of hyperedges	6
p_6:Minimum average weight of hyperedges	0.25

Table 1: Chosen parameters for our system

3.2 SemEval-2007 Results

Tables 2 and 3 show the average supervised recall, FScore, entropy and purity of our system on nouns and verbs of the test data respectively. The submitted answer consisted only of the winning cluster per instance of a target word, in effect assigning it with weight 1 (default).

Entropy measures how well the various gold standard senses are distributed within each cluster, while purity measures how pure a cluster is, containing objects from primarily one class. In general, the lower the entropy and the larger the purity values, the better the clustering algorithm performs.

Measure	Proposed methodology	MFS
Entropy	**25.5**	46.3
Purity	**89.8**	82.4
FScore	65.8	**80.7**
Sup. Recall	**81.6**	80.9

Table 2: System performance for nouns.

For nouns our system achieves a low entropy and a high purity outperforming the MFS baseline, but a lower FScore. This can be explained by the fact that the average number of clusters we produce for nouns is 11, while the gold standard average of senses is around 2.8. For verbs the performance of our system

is worse than for nouns, although entropy and purity still outperform the MFS baseline. FScore is very low, despite that the average number of clusters we produce for verbs (around 8) is less than the number of clusters we produce for nouns. This means that for verbs the senses of gold standard are much more spread among induced clusters than for nouns, causing a low unsupervised recall. Overall, FScore results are in accordance with the idea of microsenses mentioned in (Agirre et al., 2006). FScore is biased towards clusters similar to the gold standard senses and cannot capture that theory.

Measure	Proposed methodology	MFS
Entropy	**28.9**	44.4
Purity	**82.0**	77
F-score	45.1	**76.8**
Sup. Recall	73.3	**76.2**

Table 3: System performance for verbs.

Our supervised recall for verbs is 73.3%, and below the MFS baseline (76.2%), which no system managed to outperform. For nouns our supervised recall is 81.6%, which is around 0.7% above the MFS baseline. In order to fully examine the performance of our system we applied a second evaluation of our methodology using the SWSID official software.

The solution per target word instance included the entire set of clusters with their associated weights (Table 4). Results show that the submitted answer (*instance - winning_cluster*), was degrading seriously our performance both for verbs and nouns due to the loss of information in the mapping step.

POS	Proposed Methodology	MFS
Nouns	**84.3**	80.9
Verbs	75.6	**76.2**
Total	**80.2**	78.7

Table 4: Supervised recall in second evaluation.

Our supervised recall for nouns has outperformed the MFS baseline by 3.4% with the best system achieving 86.8%. Performance for verbs is 75.6%, 0.6% below the best system and MFS.

4 Conclusion

We have presented a hypergraph model for word sense induction and disambiguation. Preliminary experiments suggested that our reduced hypergraphs are small-world networks. WSI identifies the highly connected components (hubs) in the hypergraph, while WSD assigns to each cluster a score equal to the sum of weights of its hyperedges found in the local context of a target word.

Results show that our system achieves high entropy and purity performance outperforming the MFS baseline. Our methodology achieves a low FScore producing clusters that are dissimilar to the gold standard senses. Our supervised recall for nouns is 3.4% above the MFS baseline. For verbs, our supervised recall is below the MFS baseline, which no system managed to outperform.

References

Eneko Agirre and Aitor Soroa. 2007. Semeval-2007 task 2: Evaluating word sense induction and discrimination systems. In *Proceedings of SemEval-2007*. ACL.

Eneko Agirre, David Martínez, Oier López de Lacalle, and Aitor Soroa. 2006. Two graph-based algorithms for state-of-the-art wsd. In *Proceedings of the EMNLP Conference*, pages 585–593. ACL.

Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast algorithms for mining association rules in large databases. In *VLDB '94: Proceedings of the 20th International Conference on Very Large DataBases*, pages 487–499, USA. Morgan Kaufmann Publishers Inc.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The senseval-3 english lexical sample task. In *R. Mihaleca and P. Edmonds, editors, SensEval-3 Proceedings*, pages 25–28, Spain, July. ACL.

Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proceedings of CoNLL-2004*, pages 41–48. ACL.

Claudio Rocchini. 2006. Hypergraph sample image. *Wikipedia*.

Hinrich Schutze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Jean Veronis. 2004. Hyperlex:lexical cartography for information retrieval. *Computer Speech & Language*, 18(3).