Word Sense Induction Using Graphs of Collocations

Ioannis P. Klapaftis and **Suresh Manandhar** [1]

Abstract. Word Sense Induction (WSI) is the task of identifying the different senses (uses) of a target word in a given text. Traditional graph-based approaches create and then cluster a graph, in which each vertex corresponds to a word that co-occurs with the target word, and edges between vertices are weighted based on the co-occurrence frequency of their associated words. In contrast, in our approach each vertex corresponds to a collocation that co-occurs with the target word, and edges between vertices are weighted based on the co-occurrence frequency of their associated collocations. A smoothing technique is applied to identify more edges between vertices and the resulting graph is then clustered. Our evaluation under the framework of SemEval-2007 WSI task shows the following: (a) our approach produces less sense-conflating clusters than those produced by traditional graph-based approaches, (b) our approach outperforms the existing state-of-the-art results.

1 Introduction

Using word senses instead of word forms is essential in many applications such as information retrieval (IR) and machine translation (MT) [13]. Word senses are a prerequisite for word sense disambiguation (WSD) algorithms. However, they are usually represented as a fixed-list of definitions of a manually constructed lexical database. There are several disadvantages associated with the fixed-list of senses paradigm. Firstly, lexical databases often contain general definitions and miss many domain specific senses [1]. Secondly, they suffer from the lack of explicit semantic and topical relations between concepts [1]. Thirdly, they often do not reflect the exact content of the context, in which the target word appears [17]. WSI aims to overcome these limitations of hand-constructed lexicons.

Most of the work in WSI is based on the vector-space model, where each context of a target word is represented as a vector of features (e.g. frequency of co-occurring words). Context vectors are clustered and the resulting clusters are taken to represent the induced senses. Recently, graph-based methods [8, 17, 4] have been employed to WSI. Typically, graph-based approaches represent each word w_i co-occurring with the target word tw, within a pre-specified window, as a vertex. Two vertices are connected via an edge if they co-occur in one or more contexts of tw. Once the co-occurrence graph of tw has been constructed, different graph clustering algorithms are applied to induce the senses. Each cluster (induced sense) consists of a set of words that are semantically related to the particular sense.

Graph-based approaches assume that each context word, w_i, is related to one and only one sense of tw. This assumption is not always valid since w_i may appear with more than one senses of tw. For example, consider the following contexts, where the target word is *network*. The resulting graph, keeping only nouns and removing the target word is shown in Figure 1A.

- To install our satellite *system* please call our *technicians* and book an *appointment*. *Connection* to our *television* network is free of *charge*...
- To connect to the *BT* network, proceed with the *installation* of the *connection software* and then reboot your *system*...

In this example, *system* appears in two different contexts. The target word *network* appears with two different senses i.e. (1) *communication system consisting of a group of broadcasting stations (Television Network)*, and (2) *web;an interconnected system of things or people (Computer Network)*. Any hard clustering approach attempting to identify the clusters (senses) of *network* would assign *system* to only one of the two senses of *network*, even though *system* is related to both. The same problem appears for the word *connection*. Note that both *system* and *connection* are not noisy words, which could have been removed in a pre-processing stage. Instead, they are words semantically related to *network*, and hence cannot be filtered out.

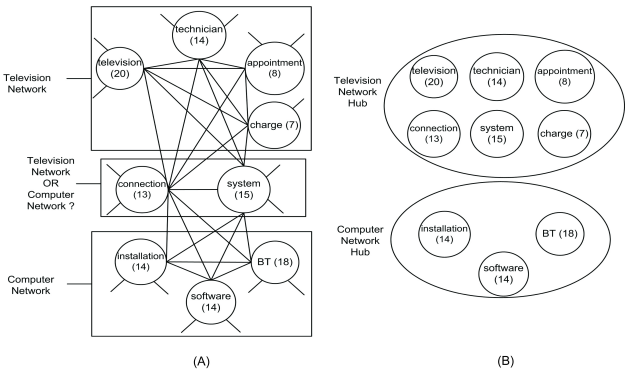

Figure 1. (A) Graph of words for the target word *network*. Numbers inside vertices correspond to their degree. (B) WSI using Véronis method [17].

The above limitation negatively affects the quality of induced senses, and tends to produce clusters conflating the target word senses. In this work we deal with this problem by creating, populating and clustering a graph, in which each vertex corresponds to a collocation[2] that co-occurs with the target word, and edges between vertices are weighted based on the co-occurrence frequency of their associated collocations.

Each produced cluster consists of a set of collocations. Our intuition is that through this tactic, we will be able to produce clusters that are less sense-conflating than those produced by current graph-based approaches, since collocations provide strong and consistent clues to

[1] Department of Computer Science, The University of York, YO10 5DD, United Kingdom, email: {*giannis,suresh*}@cs.york.ac.uk

[2] *collocation* is used in the following sense: juxtaposition of words within the same paragraph

the sense of a target word [18]. We evaluate our approach in nouns of the SemEval-2007 WSI task (SWSI) [3]. Our results confirm our intuition. The evaluation shows that our method achieves consistently high performance in all the evaluation measures outperforming the existing state-of-the-art results.

2 Related work

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) [14]. These vectors are then clustered to produce the induced senses. Figure 2 shows a simple example of four vectors taken from four contexts of the target word *network*, which appears with two different senses i.e. *Television Network* and *Computer Network*. Clustering of these vectors creates two clusters, the first one consisting of vectors *context #1* and *context #3*, and the second consisting of vectors *context #2* and *context #4*.

SenseClusters [14] is a vector-based WSI system. SenseClusters represents the contexts to be clustered using second order co-occurrence vectors. Initially, a word-by-word co-occurrence matrix is constructed by identifying bigrams that occur two or more times in the contexts to be clustered, and have a Pointwise Mutual Information (PMI) score greater than a pre-specified threshold. A row in the matrix is the co-occurrence vector for a particular context word. Each of the contexts is then represented by a single vector, which is the centroid of all the co-occurrence vectors of the words that make up the context. The k-means algorithm is used for clustering the context vectors, where the number of clusters, k, is automatically determined using the *Adapted Gap Statistic* [15].

Context #1: television, technician, cnn, system
Context #2: installation, software, BT, ADSL, computer
Context #3: television,tv, technician, cnn, nbc, cbs
Context #4: internet, ADSL, software, system, connection

Figure 2. Four context vectors for the target word *network*.

Recently, graph-based approaches [8, 17, 4] have been employed to WSI. Graph-based approaches construct a co-occurrence graph, in which words occurring in the context of the target word are represented as vertices. Two vertices share an edge, if they co-occur in the same context. Each edge receives a weight, which indicates how strong the incident vertices relate to each other.

Véronis [17] has shown that co-occurrence graphs are small-world networks and, thus, they contain highly dense subgraphs (hubs), which represent the different clusters (senses) the target word may have. To identify these hubs Véronis' algorithm iteratively finds the candidate root hub with the highest degree [3], which is then deleted along with its direct neighbours from the graph, if and only if it satisifies a set of heuristics. These heuristics are the minimum number of vertices in a hub, the average weight between the candidate root hub and its adjacent neighbours and the minimum frequency of a root hub.

For example, in Figure 1A the highest degree vertex, *television*, is the first root hub, which would be deleted along with its direct neighbours. The deleted hub corresponds to the *Television Network* sense of target word *network*. Figure 1B shows the extracted hubs following Véronis' algorithm. We observe that words *system* and *connection* have been assigned to the *Television Network* extracted hub, although they are related to the *Computer Network* hub as well.

In [8, 5], a co-occurrence graph is built for a target word by considering only nouns found in enumerations. Each noun corresponds to a vertex, and two vertices share an edge, if they co-occur in more than n enumerations. The problem of sense conflation is present here as well, since these approaches apply different hard clustering algorithms to constructed graphs.

3 Collocational graphs for WSI

Let bc, be the base corpus, which consists of paragraphs containing the target word tw. Our aim is to induce the sense of tw given bc as the only input. Let rc be a large reference corpus. In this work we have used the British National Corpus (BNC) [4].

3.1 Corpus pre-processing

Initially, tw is removed from bc and each paragraph p_i of bc and rc is POS-tagged. Following the example in [4, 2], only nouns are kept and lemmatised, since they are less ambiguous than verbs, adverbs or adjectives. At this stage each paragraph p_i both in bc and rc is a list of lemmatised nouns.

Each paragraph p_i in bc contains nouns which are semantically related to tw, as well as, common nouns which are noisy, in the sense that they are not semantically related to tw. Most graph-based WSI approaches filter out these words by applying raw frequency heuristics. In this work, we employ a more sophisticated technique based on corpora comparison using log-likelihood (G^2) [9].

Our aim is to check if the distribution of a word w_i, given it appears in bc, is similar to the distribution of w_i, given it appears in rc, i.e. $p(w_i|bc) = p(w_i|rc)$ *(null hypothesis)*. If that is true, G^2 will have a small value, and w_i should be removed from the paragraphs of bc.

$$G^2 = 2 * \sum_{i,j} n_{ij} \cdot \log\left(\frac{n_{ij}}{m_{ij}}\right) \quad (1)$$

$$m_{ij} = \frac{\sum_{k=1}^{2} n_{ik} \cdot \sum_{k=1}^{2} n_{kj}}{N} \quad (2)$$

We create two noun frequency lists. The first one, lbc, is derived from the processed bc corpus, and the second, lrc, is derived from the processed reference corpus rc. For each word $w_i \in lbc$, we create two contingency tables. The first one (OT) contains the observed counts taken from lbc and lrc (Table 1). The second (ET) contains the expected values under the model of independence (Table 2). Then we can calculate G^2 (Equation 1), where n_{ij} is the i, j cell of OT and m_{ij} (Equation 2) is the i, j cell of ET, and $N = \sum_{i,j} n_{ij}$.

Table 1. Contingency table for observed values (OT) and example for target word *network* and context word *cable*

Observed Values (OT)	Base Corpus (bc) *network*	Reference Corpus (rc) BNC
Freq. of *cable*	213 (n_{11})	2439 (n_{12})
Total Freq. of remaining words	23279 (n_{21})	24038639 (n_{22})

[3] For efficieny reason Véronis' uses the relative frequency of a vertex to identify a root hub, since the relative frequency of a vertex and its degree are linearly related.

[4] The British National Corpus (2001, version 2). Distributed by Oxford University Computing Services.

Table 2. Contingency table for expected values (ET) and example for target word *network* and context word *cable*

Expected Values (ET)	Base Corpus (bc) *network*	Reference Corpus (rc) BNC
Freq. of *cable*	2.58 (m_{11})	2649.4 (m_{12})
Total Freq. of remaining words	23489.4 (m_{21})	2.403ϵ7 (m_{22})

Following this process, we are able to identify words in lbc, which are most indicative in bc as compared to rc and vice versa. However, in this setting we are not interested in words, which have a distinctive frequency in lrc. As a result, lbc, is firstly filtered by removing words, which have a relative frequency in lbc less than in lrc. The resulting lbc is then sorted by the G^2 values. The G^2-sorted list is used to remove words from each paragraph of bc, which have a G^2 value less than a pre-specified threshold (parameter p_1). At the end of that stage, each paragraph $p_i \in bc$ is a list of nouns, which are assumed to be topically related to the target word tw. Table 3 shows the top 10 words in lbc for the target word *network*.

Table 3. Top 10 words for the target word *network*

Lemma	G^2
nbc	2319.6
cnn	1544
cable	1476.92
computer	1449.7
turner	1010.2
news	994.1
cbs	826.1
television	568.5
task	369.0
studio	438.8

3.2 Creating the initial collocational graph

A key problem at this stage is the determination of related nouns, which can be grouped into collocations, and the weighting of each such collocation. In this work, we consider collocations of size 2, i.e. they consist of two nouns. Collocations are detected by generating all the n by 2-combinations for each n-length paragraph, and then measuring their frequency. The frequency of a collocation is the number of paragraphs in the whole SWSI corpus (27132 paragraphs), which contain that collocation.

Each extracted collocation is assigned a weight, which measures the relative frequency of two nouns co-occurring. Collocations are usually weighted using information theoretic measures such as pointwise mutual information (PMI). Recently, a comparison between PMI and conditional probabilities for weighting object/verb and subject/verb pairs shows that conditional probabilities produce better results better than PMI [6], since PMI overestimates rare events. Therefore, conditional probabilities seem to be a reasonable choice for our collocation weighting.

Let freq_{ij} denote the number of paragraphs, in which nouns i, j co-occur, and freq_j denote the number of paragraphs, where noun j occurs. Then we can measure the conditional probability $p(i|j)$ using Equation 3, and $p(j|i)$ in a similar way. The final weight applied to collocation c_{ij} is the average of the calculated conditional probabilities $w_{c_{ij}} = \frac{p(i|j)+p(j|i)}{2}$.

$$p(i|j) = \frac{\text{freq}_{ij}}{\text{freq}_j} \quad (3)$$

We only extract collocations, which have frequency (parameter p_2) and weight (parameter p_3) higher than pre-specified thresholds. This filtering appears to compensate for inaccuracies in G^2, as well as for low-frequency distant collocations that are ambiguous. Each extracted and weighted collocation is represented as a vertex. Two vertices share an edge, if they co-occur, in one or more paragraphs of bc.

3.3 Weighting & populating the collocational graph

The constructed graph, G, is sparse, since we are attempting to identify rare events, i.e. edges connecting collocations. To deal with that problem, we apply a smoothing technique extending the principle that *a word is characterized by the company it keeps* [10] to collocations. Our target is both to discover new edges between vertices and to assign weights to all of the graph edges.

For each vertex i (collocation c_i), we associate a vertex vector VC_i containing the vertices (collocations), which share an edge with i in graph G. Table 4 shows an example of two vertices, i.e. *cnn_nbc* and *nbc_news*, which are not connected in G of the target word *network*.

In the next step, the similarity between each vertex vector VC_i and each vertex vector VC_j is calculated. A comparison of different similarity measures [12] shows that Jaccard similarity coefficient (JC) shows superior performance over other symmetric similarity measures such as cosine, L1 norm, euclidean distance, Jensen-Shannon divergence, etc. Therefore, we have used JC for estimating similarity between vertex vectors: $\text{JC}(VC_i, VC_j) = \frac{|\text{VC}_i \cap \text{VC}_j|}{|\text{VC}_i \cup \text{VC}_j|}$. Two collocations c_i and c_j are mutually similar if c_i is the most similar collocation to c_j and the other way round.

Two mutually similar collocations c_i and c_j are clustered with the result that an occurrence of a collocation c_k with one of c_i, c_j is also counted as an occurrence with the other collocation. For example in Table 4, if *cnn_nbc* and *nbc_news* are mutually similar, then the zero-frequency event between *nbc_news* and *cnn_tv* is set equal to the joint frequency between *cnn_nbc* and *cnn_tv*. Marginal frequencies of collocations are updated and the overall result is consequently a smoothing of relative frequencies.

Table 4. Collocations connected to *cnn_nbc* and *nbc_news*

Target:*cnn_nbc*	Target:*nbc_news*
nbc_tv	**nbc_tv**
cnn_tv	soap_opera
cnn_radio	nbc_news
news_newscast	**news_ newscast**
radio_television	nbc_newshour
cnn_headline	**cnn_headline**
nbc_politics	radio_ tv
breaking_news	**breaking_ news**

The weight applied to each edge connecting vertices i and j (collocations c_i and c_j) is the maximum of their conditional probabilities, where $p(i,j) = \frac{\text{freq}_{i,j}}{\text{freq}_j}$, $freq_i$ is the number of paragraphs collocation c_i occurs, and $p(j|i)$ is defined similarly.

3.4 Inducing senses & tagging

The final graph G', resulting from the previous stage, is clustered in order to produce the induced senses. The two criteria for choosing

a clustering algorithm were its ability to automatically induce the number of clusters and its execution time.

Markov Clustering algorithm (MCL) [7] has been extensively used in WSI [4, 8]. MCL is a fast clustering method, which is based on simulation of (stochastic) flow in graphs. The number of produced clusters depends on an inflation parameter that controls the number of produced clusters. *Chinese Whispers* (CW) [5] is a randomised graph-clustering method, time-linear to the number of edges. Contrarily to MCL, CW does not require any input parameters. However, CW is not guaranteed to converge. Evaluation of CW in WSI shows that CW performs well [5]. Biemann [5] notes that CW's ability to automatically infer the number and the size of clusters, makes it especially suited for WSI problems, where class distributions are often highly skewed and the number of classes unknown. *Normalised Min-Cut* [16] is a well-known graph-partitioning algorithm, in which a graph is partitioned in two subgraphs by minimising the total association between the two subgraphs. *Normalised Min-Cut* is iteratively applied for each extracted subgraph until a user-defined criterion is met (e.g. number of clusters). In our work, we chose to use CW for clustering our collocational graph, since compared to the above algorithms, it does not require any input parameters, it is linear to the number of edges and has already been applied to WSI.

Initially, CW assigns all vertices to different classes. Each vertex i is processed for an x (parameter p_4) number of iterations and inherits the strongest class in its local neighborhood (LN) in an update step. LN is defined as the set of vertices which share a direct connection with vertex i. During the update step for a vertex i: each class, cl, receives a score equal to the sum of the weights of edges (i, j), where j has been assigned class cl. The maximum score determines the strongest class. In case of multiple strongest classes, one is chosen randomly. Classes are updated immediately, which means that a node can inherit classes from its LN that were introduced there in the same iteration.

Word sense disambiguation (WSD) assigns one of the induced clusters to each instance of tw. Particularly, given an instance of tw in paragraph p_i: each induced cluster cl_j is assigned a score equal to the number of its collocations occurring in p_i. Our WSD exploits the one sense per collocation property [18], which means that WSD based on collocations is probably finer than WSD based on simple words, since ambiguity is reduced.

4 Evaluation

4.1 Experimental setting

We evaluate our WSI approach under the framework of SemEval-2007 WSI task (SWSI) [3]. The corpus consists of texts of the Wall Street Journal corpus, and is hand-tagged with OntoNotes senses [11]. In this paper, we focus on all 35 nouns of SWSI, ignoring verbs. We induce the senses of each target noun, tn, and then we tag each instance of tn with one of its induced senses. SWSI task organisers employ two evaluation schemes. In the first one, *unsupervised evaluation*, the results of systems are treated as clusters of target noun contexts and gold standard (GS) senses as classes. A perfect clustering solution will be the one, where each induced cluster has exactly the same contexts as one of the classes (*Homogeneity*), and each class has exactly the same contexts as one of the clusters (*Completeness*). The traditional clustering measure of F-Score is used to assess the overall quality of clustering, as well as the complementary measures of entropy and purity. Note that F-Score is a better measure than entropy or purity, since F-Score measures both homogeneity and completeness of a clustering solution, while entropy and purity measure only the first. In the second evaluation scheme, *supervised evaluation*, the training corpus is used to map the induced clusters to GS senses. The testing corpus is then used to measure the performance of systems in a WSD setting (Table 6 *Sup. Recall*).

Table 5. Chosen parameters for our approach

Parameter	Range	Value
G^2 threshold	5,10,15	$p_1 = 5$
Collocation frequency	4,6,8,10	$p_2 = 8$
Collocation weight	0.2,0.3,0.4	$p_3 = 0.2$
CW iterations	100,200	$p_4 = 200$

Our WSI methodology that uses Jaccard similarity to populate the graph is referred as *Col-JC*. *Col-BL* induces senses as *Col-JC* does, but without smoothing. We fine-tuned *Col-JC* by cross-validation in the training set of SWSI. We tried 72 combinations of parameters, and chose the setting, with the highest F-Score (Table 5). Note that SWSI participating systems *UOY*, *UBS-AC* have used labeled data for parameter estimation, while systems *I2R*, *UPV_SI*, *UMND2* do not state how their parameters were estimated [3]. *GCL* baseline is a traditional graph-based method, which builds a graph as in [17] and then uses CW to produce the clusters. The parameters of *GCL* are estimated following the process used for estimating the parameters of *Col-JC*. The *1cl1inst* baseline assigns each instance to a distinct cluster, while the *1c1w* baseline groups all instances of a target word into a single cluster. Note that the *1cl1w* baseline is equivalent to the most frequent baseline *MFS* in this setting. Tables 6 presents the unsupervised and supervised evaluation results. The fifth column in table 6 shows the average number of clusters.

4.2 Analysis of results

Evaluation of WSI methods is a difficult task. For instance, the *1cl1inst* baseline (Table 6) achieves a perfect purity and entropy. However, F-Score of *1cl1inst* is low, because senses of gold standard are spread among induced clusters causing a low unsupervised recall. Supervised recall of *1cl1inst* is undefined, due to the fact that each cluster tags one and only one instance in the corpus. Hence, clusters tagging instances in the test corpus do not tag any instances in the train corpus and the mapping cannot be performed.

Table 6. Unsupervised & supervised evaluation of WSI systems.

System	Unsupervised Evaluation				Sup.
	FSc.	Pur.	Ent.	# Cl.	Recall
UBC-AS	80.8	83.6	43.5	1.6	80.7
1c1w-MFS	80.7	82.4	46.3	1	80.9
GCL	81.1	84.0	42.7	2.3	82.3
Col-JC	78.0	88.6	31.0	5.9	86.4
Col-BL	73.1	89.6	29.0	8.0	85.6
upv_si	69.9	87.4	30.9	7.2	82.5
I2R	68.0	88.4	29.7	3.1	86.8
UMND2	67.1	85.8	37.6	1.7	84.5
UOY	65.8	89.8	25.5	11.3	81.6
1c1inst	6.6	100	0	73.1	NA

The *1c1w* baseline (Table 6) achieves high F-Score performance due to the dominance of MFS in the testing corpus. However, its purity, entropy and supervised recall are much lower than other systems, because this baseline only induces the dominant sense. *UBC-AS* seems to have a similar behaviour.

A clustering solution, which achieves high supervised recall, does not necessarily achieve high F-Score. One reason for that stems from the fact that F-Score penalises systems for getting the number of GS classes wrongly as in *1cl1inst* baseline. According to [3], supervised evaluation seems to be more neutral regarding the number of induced clusters, because clusters are mapped into a weighted vector of senses, and therefore inducing a number of clusters similar to the number of senses is not a requirement for good results. Hence, supervised recall seems to be a less biased measure for assessing WSI systems relative to the size of the training corpus. High supervised recall also means high purity and entropy as in *I2R*, but not vice versa as in *UOY*. *UOY* produces a large number of clean clusters, in effect suffering from an unreliable mapping of clusters to senses due to the lack of adequate training data. On the contrary, *I2R* produces a small number of clean clusters, in effect having a more reliable mapping and a higher supervised performance.

The above statements can be better illustrated by looking at the performance of WSI systems in both evaluation settings. Particularly, no system was able to achieve high performance in both settings, in effect being biased against one of the two evaluation schemes. However, this is not the case for our method. In Table 6, we observe that *Col-BL* (*Col-JC*) achieve 72.9% (78.0%) F-Score, outperforming the SWSI participating systems, with their entropy and purity being at high levels. In this comparison we omit the performance of *UBC-AS*, which was fine-tuned to return a number of clusters close to GS number of senses [4]. The picture is the same in the supervised evaluation, where *Col-JC* and *Col-BL* achieve high performance. Note that the performance difference between *Col-JC* and *I2R* is not statistically significant (McNemar's test at the 95% confidence level).

Our graph-based baseline, *GCL*, achieves high F-Score (81.1%), but low purity (84.0%) and high entropy (42.7%), which means that it is biased towards the *1c1w* (MFS) baseline. On the contrary, *Col-JC* achieves 88.6% purity, 31% entropy and a relatively high FScore (78.0%). The same applies for *Col-BL*, which however achieves a lower F-Score than *Col-JC*, due to the larger number of induced clusters. By examining the results on the supervised evaluation, which is a more neutral measure regarding the number of induced clusters, we observe that both *Col-JC*, *Col-BL* outperform *GCL* by a statistically significant amount. These results clearly indicate that the proposed method produces less sense conflating clusters than the traditional graph-based baseline.

The target of smoothing was to reduce the number of clusters, and obtain a better mapping of clusters to GS senses, but without affecting the clustering quality. In Table 6 we observe, that *Col-JC* has produced a smaller number of clusters than *Col-BL* with a small effect on purity and entropy. As a result, supervised recall has increased. We also observe that *Col-JC* has a higher F-Score performance than *Col-BL* due to the reduction of the number of clusters. Both *Col-BL* and *Col-JC* produce a larger number of clusters than the GS number of senses. This does not have a major effect in their F-Score performance, due to the fact that both of them generate a small number of clean large clusters, which tag the majority of instances and a higher number of small clean clusters, which tag only few instances.

5 Conclusion

We presented a graph-based WSI method, in which each vertex corresponds to a collocation that co-occurs with the target word, and edges between vertices are weighted based on the co-occurrence frequency of their associated collocations. A smoothing technique was then applied to identify more edges between vertices.

Evaluation has shown that our method produces less sense-conflating clusters than traditional graph-based approaches. Our method achieved high performance in both evaluation settings. Future work focuses on applying different collocation weighting schemes and evaluation of our approach on verbs, which are more polysemous than nouns.

REFERENCES

[1] Eneko Agirre, Olatz Ansa, David Martinez, and Eduard Hovy, 'Enriching wordnet concepts with topic signatures', in *Proceedings of the NAACL workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*. ACL, (2001).

[2] Eneko Agirre, David Martínez, Oier López de Lacalle, and Aitor Soroa, 'Two graph-based algorithms for state-of-the-art wsd', in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 585–593, Sydney, Australia, (July 2006). ACL.

[3] Eneko Agirre and Aitor Soroa, 'Semeval-2007 task 02: Evaluating word sense induction and discrimination systems', in *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pp. 7–12, Prague, Czech Republic, (June 2007). ACL.

[4] Eneko Agirre and Aitor Soroa, 'Ubc-as: A graph based unsupervised system for induction and classification', in *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pp. 346–349, Prague, Czech Republic, (June 2007). ACL.

[5] Chris Biemann, 'Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems', in *Proceedings of TextGraphs*, pp. 73–80, New York,USA, (June 2006). ACL.

[6] Philipp Cimiano, Andreas Hotho, and Steffen Staab, 'Learning concept hierarchies from text corpora using formal concept analysis', *Journal of Artificial Intelligence Research (JAIR)*, **24**, 305–339, (2005).

[7] Stijn Dongen, 'Performance criteria for graph clustering and markov cluster experiments', Technical report, CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, (2000).

[8] Beate Dorow and Dominic Widdows, 'Discovering corpus-specific word senses', in *Proceedings of the 10th conference of the European chapter of the ACL*, pp. 79–82, Budapest, Hungary, (2003). ACL.

[9] Ted Dunning, 'Accurate methods for the statistics of surprise and coincidence', *Comput. Linguist.*, **19**(1), 61–74, (1993).

[10] R. Firth, John, 'A synopsis of linguistic theory, 1930-1955', in *Studies in Linguistic Analysis*, 1–32, Basic Blackwell, Oxford, 1st edn., (1957). Special Volume of the Philological Society.

[11] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel, 'Ontonotes: The 90% solution', in *Proceedings of the Human Language Technology / North American Association for Computational Linguistics conference*, New York, USA, (2006).

[12] Lillian Lee, 'Measures of distributional similarity', in *37th Annual Meeting of the Association for Computational Linguistics*, pp. 25–32, Maryland,USA, (1999). ACL.

[13] Patrick Pantel and Dekang Lin, 'Discovering word senses from text', in *KDD '02: Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 613–619, New York, NY, USA, (2002). ACM.

[14] Ted Pedersen, 'Umnd2 : Senseclusters applied to the sense induction task of senseval-4', in *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pp. 394–397, Prague, Czech Republic, (June 2007). ACL.

[15] Ted Pedersen and Anagha Kulkarni, 'Selecting the "right" number of senses based on clustering criterion functions', in *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, (2006). ACL.

[16] Jianbo Shi and Jitendra Malik, 'Normalized cuts and image segmentation', in *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, p. 731, Washington, USA, (1997). IEEE Computer Society.

[17] Jean Véronis, 'Hyperlex: lexical cartography for information retrieval', *Computer Speech & Language*, **18**(3), 223–252, (2004).

[18] David Yarowsky, 'Unsupervised word sense disambiguation rivaling supervised methods', in *Proceedings of the 33rd annual meeting of the Association for Computational Linguistics*, pp. 189–196, Massachusetts,USA, (1995). ACL.