Two graph-based algorithms for state-of-the-art WSD

Eneko Agirre, David Martínez, Oier López de Lacalle and Aitor Soroa
IXA NLP Group
University of the Basque Country
Donostia, Basque Contry
a.soroa@si.ehu.es

Abstract

This paper explores the use of two graph algorithms for unsupervised induction and tagging of nominal word senses based on corpora. Our main contribution is the optimization of the free parameters of those algorithms and its evaluation against publicly available gold standards. We present a thorough evaluation comprising supervised and unsupervised modes, and both lexical-sample and all-words tasks. The results show that, in spite of the information loss inherent to mapping the induced senses to the gold-standard, the optimization of parameters based on a small sample of nouns carries over to all nouns, performing close to supervised systems in the lexical sample task and yielding the second-best WSD systems for the Senseval-3 all-words task.

1 Introduction

Word sense disambiguation (WSD) is a key enabling-technology. Supervised WSD techniques are the best performing in public evaluations, but need large amounts of hand-tagged data. Existing hand-annotated corpora like SemCor (Miller et al., 1993), which is annotated with WordNet senses (Fellbaum, 1998) allow for a small improvement over the simple most frequent sense heuristic, as attested in the all-words track of the last Senseval competition (Snyder and Palmer, 2004). In theory, larger amounts of training data (SemCor has approx. 700K words) would improve the performance of supervised WSD, but no current project exists to provide such an expensive resource.

Supervised WSD is based on the "fixed-list of senses" paradigm, where the senses for a target word are a closed list coming from a dictionary or lexicon. Lexicographers and semanticists have long warned about the problems of such an approach, where senses are listed separately as discrete entities, and have argued in favor of more complex representations, where, for instance, senses are dense regions in a continuum (Cruse, 2000).

Unsupervised WSD has followed this line of thinking, and tries to induce word senses directly from the corpus. Typical unsupervised WSD systems involve clustering techniques, which group together similar examples. Given a set of induced clusters (which represent word *uses* or senses[1]), each new occurrence of the target word will be compared to the clusters and the most similar cluster will be selected as its sense.

Most of the unsupervised WSD work has been based on the vector space model, where each example is represented by a vector of features (e.g. the words occurring in the context), and the induced senses are either clusters of examples (Schütze, 1998; Purandare and Pedersen, 2004) or clusters of words (Pantel and Lin, 2002). Recently, Véronis (Véronis, 2004) has proposed HyperLex, an application of graph models to WSD based on the small-world properties of cooccurrence graphs. Graph-based methods have gained attention in several areas of NLP, including knowledge-based WSD (Mihalcea, 2005; Navigli and Velardi, 2005) and summarization (Erkan and Radev, 2004; Mihalcea and Tarau, 2004).

The HyperLex algorithm presented in (Véronis, 2004) is entirely corpus-based. It builds a cooccurrence graph for all pairs of words cooccurring in the context of the target word. Véronis shows that this kind of graph fulfills the properties of small world graphs, and thus possesses highly connected

[1] Unsupervised WSD approaches prefer the term 'word uses' to 'word senses'. In this paper we use them interchangeably to refer to both the induced clusters, and to the word senses from some reference lexicon.

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 585–593,
Sydney, July 2006. ©2006 Association for Computational Linguistics

components (hubs) in the graph. These hubs eventually identify the main word uses (senses) of the target word, and can be used to perform word sense disambiguation. These hubs are used as a representation of the senses induced by the system, the same way that clusters of examples are used to represent senses in clustering approaches to WSD (Purandare and Pedersen, 2004).

One of the problems of unsupervised systems is that of managing to do a fair evaluation. Most of current unsupervised systems are evaluated in-house, with a brief comparison to a re-implementation of a former system, leading to a proliferation of unsupervised systems with little ground to compare among them.

In preliminary work (Agirre et al., 2006), we have shown that HyperLex compares favorably to other unsupervised systems. We defined a semi-supervised setting for optimizing the free-parameters of HyperLex on the Senseval-2 English Lexical Sample task (S2LS), which consisted on mapping the induced senses onto the official sense inventory using the training part of S2LS. The best parameters were then used on the Senseval-3 English Lexical Sample task (S3LS), where a similar semi-supervised method was used to output the official sense inventory.

This paper extends the previous work in several aspects. First of all, we adapted the PageRank graph-based method (Brin and Page, 1998) for WSD and compared it with HyperLex.

We also extend the previous evaluation scheme, using measures in the clustering community which only require a gold standard clustering and no mapping step. This allows for having a purely unsupervised WSD system, and at the same time comparing supervised and unsupervised systems according to clustering criteria.

We also include the Senseval-3 English All-words testbed (S3AW), where, in principle, unsupervised and semi-supervised systems have an advantage over purely supervised systems due to the scarcity of training data. We show that our system is competitive with supervised systems, ranking second.

This paper is structured as follows. We first present two graph-based algorithms, HyperLex and PageRank. Section 3 presents the two evaluation frameworks. Section 4 introduces parameter optimization. Section 5 shows the experimental setting and results. Section 6 analyzes the results and presents related work. Finally, we draw the conclusions and advance future work.

2 A graph algorithm for corpus-based WSD

The basic steps for our implementation of HyperLex and its variant using PageRank are common. We first build the cooccurrence graph, then we select the hubs that are going to represent the senses using two different strategies inspired by HyperLex and PageRank. We are then ready to use the induced senses to do word sense disambiguation.

2.1 Building cooccurrence graphs

For each word to be disambiguated, a text corpus is collected, consisting of the paragraphs where the word occurs. From this corpus, a cooccurrence graph for the target word is built. Vertices in the graph correspond to words[2] in the text (except the target word itself). Two words appearing in the same paragraph are said to cooccur, and are connected with edges. Each edge is assigned a weight which measures the relative frequency of the two words cooccurring. Specifically, let w_{ij} be the weight of the edge[3] connecting nodes i and j, then $w_{ij} = 1 - \max[P(i \mid j), P(j \mid i)]$, where $P(i \mid j) = \frac{\text{freq}_{ij}}{\text{freq}_j}$ and $P(j \mid i) = \frac{\text{freq}_{ij}}{\text{freq}_i}$.

The weight of an edge measures how tightly connected the two words are. Words which always occur together receive a weight of 0. Words rarely cooccurring receive weights close to 1.

2.2 Selecting hubs: HyperLex vs. PageRank

Once the cooccurrence graph is built, Véronis proposes a simple iterative algorithm to obtain its hubs. At each step, the algorithm finds the vertex with highest relative frequency[4] in the graph, and, if it meets some criteria, it is selected as a hub. These criteria are determined by a set of heuristic parameters, that will be explained later in Section 4. After a vertex is selected to be a hub, its neighbors are no longer eligible as hub candidates. At any time, if the next vertex candidate has a relative frequency below a certain threshold, the algorithm stops.

Another alternative is to use the PageRank algorithm (Brin and Page, 1998) for finding hubs in the

[2] Following Véronis, we only work on nouns.

[3] The cooccurrence graph is undirected, i.e. $w_{ij} = w_{ji}$

[4] In cooccurrence graphs, the relative frequency of a vertex and its degree are linearly related, and it is therefore possible to avoid the costly computation of the degree.

coocurrence graph. PageRank is an iterative algorithm that ranks all the vertices according to their relative importance within the graph following a random-walk model. In this model, a link between vertices v_1 and v_2 means that v_1 *recommends* v_2. The more vertices recommend v_2, the higher the rank of v_2 will be. Furthermore, the rank of a vertex depends not only on how many vertices point to it, but on the rank of these vertices as well.

Although PageRank was initially designed to work with directed graphs, and with no weights in links, the algorithm can be easily extended to model undirected graphs whose edges are weighted. Specifically, let $G = (V, E)$ be an undirected graph with the set of vertices V and set of edges E. For a given vertex v_i, let $\mathrm{In}(v_i)$ be the set of vertices pointing to it[5]. The rank of v_i is defined as:

$$P(v_i) = (1 - d) + d \sum_{j \in \mathrm{In}(v_i)} \frac{w_{ji}}{\sum_{k \in \mathrm{In}(v_j)} w_{jk}} P(v_j)$$

where w_{ij} is the weight of the link between vertices v_i and v_j, and $0 \leq d \leq 1$. d is called the *damping factor* and models the probability of a web surfer standing at a vertex to follow a link from this vertex (probability d) or to jump to a random vertex in the graph (probability $1 - d$). The factor is usually set at 0.85.

The algorithm initializes the ranks of the vertices with a fixed value (usually $\frac{1}{N}$ for a graph with N vertices) and iterates until convergence below a given threshold is achieved, or, more typically, until a fixed number of iterations are executed. Note that the convergence of the algorithms doesn't depend on the initial value of the ranks.

After running the algorithm, the vertices of the graph are ordered in decreasing order according to its rank, and a number of them are chosen as the main hubs of the word. The hubs finally selected depend again of some heuristics and will be described in section 4.

2.3 Using hubs for WSD

Once the hubs that represent the senses of the word are selected (following any of the methods presented in the last section), each of them is linked to the target word with edges weighting 0, and the *Minimum Spanning Tree* (MST) of the whole graph is calculated and stored.

The MST is then used to perform word sense disambiguation, in the following way. For every instance of the target word, the words surrounding it are examined and looked up in the MST. By construction of the MST, words in it are placed under exactly one hub. Each word in the context receives a set of scores s, with one score per hub, where all scores are 0 except the one corresponding to the hub where it is placed. If the scores are organized in a score vector, all values are 0, except, say, the i-th component, which receives a score $d(h_i, v)$, which is the distance between the hub h_i and the node representing the word v. Thus, $d(h_i, v)$ assigns a score of 1 to hubs and the score decreases as the nodes move away from the hub in the tree.

For a given occurrence of the target word, the score vectors of all the words in the context are added, and the hub that receives the maximum score is chosen.

3 Evaluating unsupervised WSD systems

All unsupervised WSD algorithms need some addition in order to be evaluated. One alternative, as in (Véronis, 2004), is to manually decide the correctness of the hubs assigned to each occurrence of the words. This approach has two main disadvantages. First, it is expensive to manually verify each occurrence of the word, and different runs of the algorithm need to be evaluated in turn. Second, it is not an easy task to manually decide if an occurrence of a word effectively corresponds with the use of the word the assigned hub refers to, specially considering that the person is given a short list of words linked to the hub. Besides, it is widely acknowledged that people are leaned not to contradict the proposed answer.

A second alternative is to evaluate the system according to some performance in an application, e.g. information retrieval (Schütze, 1998). This is a very attractive idea, but requires expensive system development and it is sometimes difficult to separate the reasons for the good (or bad) performance.

A third alternative would be to devise a method to map the hubs (clusters) returned by the system to the senses in a lexicon. Pantel and Lin (2002) automatically mapped the senses to WordNet, and then measured the quality of the mapping. More recently, tagged corpora have been used to map the induced senses, and then compare the systems over publicly available benchmarks (Puran-

[5] As G is undirected, the in-degree of a vertex v is equal to its out-degree.

dare and Pedersen, 2004; Niu et al., 2005; Agirre et al., 2006), which offers the advantage of comparing to other systems, but converts the whole system into semi-supervised. See Section 5 for more details on these systems. Note that the mapping introduces noise and information loss, which is a disadvantage when comparing to other systems that rely on the gold-standard senses.

Yet another possibility is to evaluate the induced senses against a gold standard as a clustering task. Induced senses are clusters, gold standard senses are classes, and measures from the clustering literature like entropy or purity can be used. In this case the manually tagged corpus is taken to be the gold standard, where a class is the set of examples tagged with a sense.

We decided to adopt the last two alternatives, since they allow for comparison over publicly available systems of any kind.

3.1 Evaluation of clustering: hubs as clusters

In this setting the selected hubs are treated as clusters of examples and gold standard senses are classes. In order to compare the clusters with the classes, hand annotated corpora are needed (for instance Senseval). The test set is first tagged with the induced senses. A perfect clustering solution will be the one where each cluster has exactly the same examples as one of the classes, and vice versa. The evaluation is completely unsupervised.

Following standard cluster evaluation practice (Zhao and Karypis, 2005), we consider three measures: entropy, purity and Fscore. The entropy measure considers how the various classes of objects are distributed within each cluster. In general, the smaller the entropy value, the better the clustering algorithm performs. The purity measure considers the extent to which each cluster contained objects from primarily one class. The larger the values of purity, the better the clustering algorithm performs. The Fscore is used in a similar fashion to Information Retrieval exercises, with precision and recall defined as the percentage of correctly "retrieved" examples for a cluster (divided by total cluster size), and recall as the percentage of correctly "retrieved" examples for a cluster (divided by total class size). For a formal definition refer to (Zhao and Karypis, 2005). If the clustering is identical to the original classes in the datasets, FScore will be equal to one which means that the higher the FScore, the better the clustering is.

3.2 Evaluation as supervised WSD: mapping hubs to senses

(Agirre et al., 2006) presents a straightforward framework that uses hand-tagged material in order to map the induced senses into the senses used in a gold standard . The WSD system first tags the training part of some hand-annotated corpus with the induced hubs. The hand labels are then used to construct a matrix relating assigned hubs to existing senses, simply counting the times an occurrence with sense s_j has been assigned hub h_i. In the testing step we apply the WSD algorithm over the test corpus, using the hubs-to-senses matrix to select the sense with highest weights. See (Agirre et al., 2006) for further details.

4 Tuning the parameters

The behavior of the original HyperLex algorithm was influenced by a set of heuristic parameters, which were set by Véronis following his intuition. In (Agirre et al., 2006) we tuned the parameters using the mapping strategy for evaluation. We set a range for each of the parameters, and evaluated the algorithm for each combination of the parameters on a fixed set of words (S2LS), which was different from the final test sets (S3LS and S3AW). This ensures that the chosen parameter set can be used for any noun, and is not overfitted to a small set of nouns.

In this paper, we perform the parameter tuning according to four different criteria, i.e., best supervised performance and best unsupervised entropy/purity/FScore performance. At the end, we have four sets of parameters (those that obtained the best results in S2LS for each criterion), and each set is then selected to be run against the S3LS and S3AW datasets.

The parameters of the graph-based algorithm can be divided in two sets: those that affect how the cooccurrence graph is built (p1–p4 below), and those that control the way the hubs are extracted from it (p5–p8 below).

p1	Minimum frequency of edges (occurrences)
p2	Minimum frequency of vertices (words)
p3	Edges with weights above this value are removed
p4	Context containing fewer words are not processed
p5	Minimum number of adjacent vertices in a hub
p6	Max. mean weight of the adjacent vertices of a hub
p7	Minimum frequency of hubs
p8	Number of selected hubs

		Vr_opt		Pr_fr (p7) and Pr_fx (p8)	
	Vr	Range	Best	Range	Best
p1	5	1-3	1	1-3	2
p2	10	2-4	3	2-4	3
p3	.9	.3-.7	.4	.4-.5	.5
p4	4	4	4	4	4
p5	6	1-7	1	–	–
p6	.8	.6-.95	.95	–	–
p7	.001	.0009-.003	.001	.0015-.0025	.0016
p8	–	–	–	50-65	55

Table 1: Parameters of the HyperLex algorithm

Both strategies to select hubs from the coocurrence graph (cf. Sect. 2.2) share parameters p1–p4. The algorithm proposed by Véronis uses p5–p6 as requirements for hubs, and p7 as the threshold to stop looking for more hubs: candidates with frequency below p7 are not eligible to be hubs.

Regarding PageRank the original formulation does not have any provision for determining which are hubs and which not, it just returns a weighted list of vertices. We have experimented with two methods: a threshold for the frequency of the hubs (as before, p7), and a fixed number of hubs for every target word (p8). For a shorthand we use Vr for Veronis' original formulation with default parameters, Vr_opt for optimized parameters, and Pr_fr and Pr_fx respectively for the two ways of using PageRank.

Table 1 lists the parameters of the HyperLex algorithm, with the default values proposed for them in the original work (second column), the ranges that we explored, and the optimal values according to the supervised recall evaluation (cf. Sect. 3.1). For Vr_opt we tried 6700 combinations. PageRank has less parameters, and we also used the previous optimization of Vr_opt to limit the range of p4, so Pr_fr and Pr_fx get respectively 180 and 288 combinations.

5 Experiment setting and results

To evaluate the HyperLex algorithm in a standard benchmark, we will first focus on a more extensive evaluation of S3LS and then see the results in S3AW (cf. Sec. 5.4). Following the design for evaluation explained in Section 3, we use the standard train-test split for the supervised evaluation, while the unsupervised evaluation only uses the test part.

	Sup.	Unsupervised		
	Rec.	Entr.	Pur.	FS
Vr	59.9	50.3	58.2	**44.1**
Vr_opt	**64.6**	**18.3**	**78.5**	35.0
Pr_fr	64.5	18.7	77.2	34.3
Pr_fx	62.2	25.4	72.2	33.3
1ex-1hub	40.1	0.0	100.0	14.5
MFS	54.5	53.2	52.8	28.3
S3LS-best	**72.9**	**19.9**	**67.3**	**63.8**
kNN-all	70.6	21.2	64.0	60.6
kNN-BoW	63.5	22.6	61.1	57.1
Cymfony (10%-S3LS)	**57.9**	**25.0**	**55.7**	**52.0**
Prob0 (MFS-S3)	54.2	28.8	49.3	46.0
clr04 (MFS-Sc)	48.8	25.8	52.5	46.2
Ciaosenso (MFS-Sc)	48.7	28.0	50.3	48.8
duluth-senserelate	47.5	27.2	51.1	44.9

Table 2: Results for the nouns in S3LS using the 4 methods (Vr, Vr_opt, Pr_fr and Pr_fx). Each of the methods was optimized in S2LS using the 4 evaluation criteria (Supervised recall, Entropy, Purity and Fscore) and evaluated on S3LS according to the respective evaluation criteria (in the columns). Two baselines, plus 3 supervised and 5 unsupervised systems are also shown. Bold is used for best results in each category.

Table 2 shows the results of the 4 variants of our algorithm. Vr stands for the original Veronis algorithm with default parameters, Vr_opt to our optimized version, and Pr_fr and Pr_fx to the two variants of PageRank. In the columns we find the evaluation results according to our 4 criteria. For supervised evaluation we indicate only recall, which in our case equals precision, as the coverage is 100% in all cases (values returned by the official Senseval scorer). We also include 2 baselines, a system returning a single cluster (that of the most frequent sense, MFS), and another returning one cluster for each example (1ex-1hub). The last rows list the results for 3 supervised and 5 unsupervised systems (see Sect. 5.1). We will comment on the result of this table from different perspectives.

5.1 Supervised evaluation

In this subsection we will focus in the first four evaluation rows in Table 2. All variants of the algorithm outperform by an ample margin the MFS and the 1ex-1hub baselines when evaluated on S3LS recall. This means that the method is able to learn useful hubs. Note that we perform this supervised evaluation just for comparison with other systems, and to prove that we are able to provide high performance WSD.

The default parameter setting (Vr) gets the worst results, followed by the fixed-hub implementation of PageRank (Pr_fx). Pagerank with frequency threshold (Pr_fr) and the optimized Veronis (Vr_opt) obtain a 10 point improvement over the MFS baseline with very similar results (the difference is not statistically significant according to McNemar's test at 95% confidence level).

Table 2 also shows the results of three supervised systems. These results (and those of the other unsupervised systems in the table) where obtained from the Senseval website, and the only processing we did was to filter nouns. S3LS-best stands for the the winner of S3LS (Mihalcea et al., 2004), which is 8.3 points over our method. We also include the results of two of our in-house systems. kNN-all is a state-of-the-art system (Agirre et al., 2005) using wide range of local and topical features, and only 2.3 points below the best S3LS system. kNN-BoW which is the same supervised system, but restricted to bag-of-words features only, which are the ones used by our graph-based systems. The table shows that Vr_opt and Pr_fr are one single point from kNN-BoW, which is an impressive result if we take into account the information loss of the mapping step and that we tuned our parameters on a different set of words.

The last 5 rows of Table 2 show several unsupervised systems, all of which except Cymfony (Niu et al., 2005) and (Purandare and Pedersen, 2004) participated in S3LS (check (Mihalcea et al., 2004) for further details on the systems). We classify them according to the amount of "supervision" they have: some have access to most-frequent information (MFS-S3 if counted over S3LS, MFS-Sc if counted over SemCor), some use 10% of the S3LS training part for mapping (10%-S3LS). Only one system (Duluth) did not use in any way hand-tagged corpora.

The table shows that Vr_opt and Pr_fr are more than 6 points above the other unsupervised systems, but given the different typology of unsupervised systems, it's unfair to draw definitive conclusions from a raw comparison of results. The system coming closer to ours is that described in (Niu et al., 2005). They use hand tagged corpora which does not need to include the target word to tune the parameters of a rather complex clustering method which does use local features. They do use the S3LS training corpus for mapping. For every sense of the target word, three of its contexts in the train corpus are gathered (around 10% of the training data) and tagged. Each cluster is then related with its most frequent sense. The mapping method is similar to ours, but we use all the available training data and allow for different hubs to be assigned to the same sense.

Another system similar to ours is (Purandare and Pedersen, 2004), which unfortunately was evaluated on Senseval 2 data and is not included in the table. The authors use first and second order bag-of-word context features to represent each instance of the corpus. They apply several clustering algorithms based on the vector space model, limiting the number of clusters to 7. They also use all available training data for mapping, but given their small number of clusters they opt for a one-to-one mapping which maximizes the assignment and discards the less frequent clusters. They also discard some difficult cases, like senses and words with low frequencies (10% of total occurrences and 90, respectively). The different test set and mapping system make the comparison difficult, but the fact that the best of their combinations beats MFS by 1 point on average (47.6% vs. 46.4%) for the selected nouns and senses make us think that our results are more robust (nearly 10% over MFS).

5.2 Clustering evaluation

The three columns corresponding to fully unsupervised evaluation in Table 2 show that all our 3 optimized variants easily outperform the MFS baseline. The best results are in this case for the optimized Veronis, followed closely by Pagerank with frequency threshold.

The comparison with the supervised and unsupervised systems shows that our system gets better entropy and purity values, but worse FScore. This can be explained by the bias of entropy and purity towards smaller and more numerous clusters. In fact the 1ex-1hub baseline obtains the best entropy and purity scores. Our graph-based system tends to induce a large number of senses (with averages of 60 to 70 senses). On the other hand FScore penalizes the systems inducing a different number of clusters. As the supervised and unsupervised systems were designed to return the same (or similar) number of senses as in the gold standard, they attain higher FScores. This motivated us to compare the results of the best parameters across evaluation methods.

5.3 Comparison across evaluation methods

Table 3 shows all 16 evaluation possibilities for each variant of the algorithm, depending of the evaluation criteria used in S2LS (in the rows) and the evaluation criteria used in S3LS (in the columns). This table shows that the best results (in bold for each variant) tend to be in the diagonal, that is, when the same evaluation criterion is used for optimization and test, but it is not decisive. If we take the first row (supervised evaluation) as the most credible criterion, we can see that optimizing according to entropy and purity get similar and sometimes better result (Pr_fr and Pr_fx). On the contrary the Fscore yields worse results by far.

This indicates that a purely unsupervised system evaluated according to the gold standard (based on entropy or purity) yields optimal parameters similar to the supervised (mapped) version. This is an important result, as it shows that the quality in performance does not come from the mapping step, but from the algorithm and optimal parameter setting. The table shows that optimization on purity and entropy criteria do correlate with good performance in the supervised evaluation.

The failure of FScore based optimization, in our opinion, indicates that our clustering algorithm prefers smaller and more numerous clusters, compared to the gold standard. FScore prefers clustering solutions that have a similar number of clusters to that of the gold standard, but it is unable to drive the optimization or our algorithm towards good results in the supervised evaluation.

All in all, the best results are attained with smaller and more numerous hubs, a kind of microsenses. This effect is the same for all three variants tried and all evaluation criteria, with Fscore yielding less clusters. At first we were uncomfortable with this behavior, so we checked whether HyperLex was degenerating into a trivial solution. This was the main reason to include the 1ex-1hub baseline, which simulates a clustering algorithm returning one hub per example, and its precision was 40.1, well below the MFS baseline. We also realized that our results are in accordance with some theories of word meaning, e.g. the "indefinitely large set of prototypes-within-prototypes" envisioned in (Cruse, 2000). Ted Pedersen has also observed a similar behaviour in his vectorspace model clustering experiments (PC). We now think that the idea of having many micro-senses is attractive for further exploration, specially if we are able to organize them into coarser hubs in future work.

		Sup.	Unsupervised		
Alg.	Opt.	Rec.	Entr.	Pur.	FS
Vr	Sup	**64.6**	18.4	77.9	30.0
	Ent	**64.6**	**18.3**	78.3	29.1
	Pur	63.7	19.0	**78.5**	30.8
	Fsc	60.4	38.2	63.5	**35.0**
Pr_fr	Sup	64.5	20.8	76.1	28.6
	Ent	64.6	**18.7**	**77.7**	27.2
	Pur	**64.7**	19.3	77.2	27.6
	Fsc	61.2	36.0	65.2	**34.3**
Pr_fx	Sup	62.2	28.2	69.3	29.5
	Ent	**63.1**	**25.4**	**72.2**	28.4
	Pur	**63.1**	**25.4**	**72.2**	28.4
	Fsc	54.5	32.9	66.5	**33.3**

Table 3: Cross-evaluation comparison. In the rows the evaluation method for optmizing over S2LS is shown, and in the columns the result over S3LS according to the different evaluation methods.

	recall
kuaw	70.9
Pr_fr	70.7
Vr_opt	70.1
GAMBL	70.1
MFS	69.9
LCCaw	68.6

Table 4: Results for the nouns in S3AW, compared to the most frequent baseline and the top three supervised systems

5.4 S3AW task

In the Senseval-3 all-words task (Snyder and Palmer, 2004) all words in three document excerpts need to be disambiguated. Given the scarce amount of training data available in Semcor (Miller et al., 1993), supervised systems barely improve upon the simple most frequent heuristic. In this setting the unsupervised evaluation schemes are not feasible, as many of the target words occur only once, so we used the mapping strategy with Semcor to produce the required WordNet senses in the output.

Table 4 shows the results for our systems with the best parameters according to the supervised criterion on S2LS, plus the top three S3AW supervised systems and the most frequent sense heuristic. In order to focus the comparison, we only kept noun occurrences of all systems and filtered out multiwords, target words with two different lemmas and unknown tags, leaving a total of 857 occurrences of nouns. We can see that Pr_fr is only 0.2 from the S3AW winning system, demonstrating that our unsupervised graph-based systems that use Semcor for mapping are nearly equivalent to the most powerful supervised systems to date. In fact, the differences in performance for the systems are not statistically significant (McNemar's test at 95% significance level).

6 Conclusions and further work

This paper has explored the use of two graph algorithms for corpus-based disambiguation of nominal senses. We have shown that the parameter optimization learnt over a small set of nouns significantly improves the performance for all nouns, and produces a system which (1) in a lexical-sample setting (Senseval 3 dataset) is 10 points over the Most-Frequent-Sense baseline, 1 point over a supervised system using the same kind of information (i.e. bag-of-words features), and 8 points below the best supervised system, and (2) in the all-words setting is à la par the best supervised system. The performance of PageRank is statistically the same as that of HyperLex, with the advantage of PageRank of using less parameters.

In order to compete on the same test set as supervised systems, we do use hand-tagged data, but only to do the mapping from the induced senses into the gold standard senses. In fact, we believe that using our WSD system as a purely unsupervised system (i.e. returning just hubs), the perfomance would be higher, as we would avoid the information loss in the mapping. We would like to test this on Information Retrieval, perhaps on a setting similar to that of (Schütze, 1998), which would allow for an indirect evaluation of the quality and a comparison with supervised WSD system on the same grounds.

We have also shown that the optimization according to purity and entropy values (which does not need the supervised mapping step) yields very good parameters, comparable to those obtained in the supervised optimization strategy. This indicates that we are able to optimize the algorithm in a completely unsupervised fashion for a small number of words, and then carry over to tag new text with the induced senses.

Regarding efficiency, our implementation of HyperLex is extremely fast. Trying the 6700 combinations of parameters takes 5 hours in a 2 AMD Opteron processors at 2GHz and 3Gb RAM. A single run (building the MST, mapping and tagging the test sentences) takes only 16 sec. For this reason, even if an on-line version would be in principle desirable, we think that this batch version is readily usable as a standalone word sense disambiguation system.

Both graph-based methods and vector-based clustering methods rely on local information, typically obtained by the occurrences of neighbor words in context. The advantage of graph-based techniques over over vector-based clustering might come from the fact that the former are able to measure the relative importance of a vertex in the whole graph, and thus combine both local and global cooccurrence information.

For the future, we would like to look more closely the micro-senses induced by HyperLex, and see if we can group them into coarser clusters. We would also like to integrate different kinds of information, specially the local or syntactic features so successfully used by supervised systems, but also more heterogeneous information from knowledge bases.

Graph models have been very successful in some settings (e.g. the PageRank algorithm of Google), and have been rediscovered recently for natural language tasks like knowledge-based WSD, textual entailment, summarization and dependency parsing. Now that we have set a robust optimization and evaluation framework we would like to test other such algorithms (e.g. HITS (Kleinberg, 1999)) in the same conditions.

Acknowledgements

Oier Lopez de Lacalle enjoys a PhD grant from the Basque Government. We thank the comments of the three anonymous reviewers.

References

E. Agirre, O. Lopez de Lacalle, and D. Martinez. 2005. Exploring feature spaces with svd and unlabeled data for word sense disambiguation. In *Proc. of RANLP*.

E. Agirre, O. Lopez de Lacalle, D. Martinez, and A. Soroa. 2006. Evaluating and optimizing the parameters of an unsupervised graph-based wsd algorithm. In *Proc. of the NAACL Texgraphs workshop*.

S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7).

D. A. Cruse, 2000. *Polysemy: Theoretical and Computational Approaches*, chapter Aspects of the Microstructure of Word Meanings, pages 31–51. OUP.

G Erkan and D. R. Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

R. Mihalcea and P Tarau. 2004. Textrank: Bringing order into texts. In *Proc. of EMNLP2004*.

R. Mihalcea, T. Chklovski, and A. Kilgarriff. 2004. The senseval-3 english lexical sample task. In R. Mihalcea and P. Edmonds, editors, *Senseval-3 proccedings*, pages 25–28. ACL, July.

R. Mihalcea. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proc. of EMNLP2005*.

G.A. Miller, C. Leacock, R. Tengi, and R.Bunker. 1993. A semantic concordance. In *Proc. of the ARPA HLT workshop*.

R. Navigli and P. Velardi. 2005. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(27):1063–1074, June.

C. Niu, W. Li, R. K. Srihari, and H. Li. 2005. Word independent context pair classification model for word sense disambiguation. In *Proc. of CoNLL-2005*.

P. Pantel and D. Lin. 2002. Discovering word senses from text. In *Proc. of KDD02*.

A. Purandare and T. Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *Proc. of CoNLL-2004*, pages 41–48.

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

B. Snyder and M. Palmer. 2004. The english all-words task. In *Proc. of SENSEVAL*.

J. Véronis. 2004. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252.

Y Zhao and G Karypis. 2005. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168.