Merging Word Senses

Sumit Bhagwani, Shrutiranjan Satapathy, Harish Karnick
Computer Science and Engineering
IIT Kanpur, Kanpur - 208016, India
{sumitb,sranjans,hk}@cse.iitk.ac.in

Abstract

WordNet, a widely used sense inventory for Word Sense Disambiguation(WSD), is often too fine-grained for many Natural Language applications because of its narrow sense distinctions. We present a semi-supervised approach to learn similarity between WordNet synsets using a graph based recursive similarity definition. We seed our framework with sense similarities of all the word-sense pairs, learnt using supervision on human-labelled sense clusterings. Finally we discuss our method to derive coarse sense inventories at arbitrary granularities and show that the coarse-grained sense inventory obtained significantly boosts the disambiguation of nouns on standard test sets.

1 Introduction

With different applications requiring different levels of word sense granularity, producing sense clustered inventories with the requisite level of sense granularity has become important. The subtleties of sense distinctions captured by WordNet(Miller, 1995) are helpful for language learners (Snow et al., 2007) and in machine translation of languages as diverse as Chinese and English (Ng et al., 2003). On the other hand, for tasks like Document Categorization and Information Retrieval (Buitelaar, 2000), it may be sufficient to know if a given word belongs to a coarsely defined class of WordNet senses. Using the fine grained sense inventory of WordNet may be detrimental to the performance of these applications. Thus developing a framework which can generate sense inventories with different granularities can improve the performance of many applications.

To generate a coarse sense inventory, many researchers have focused on generating coarse senses for each word by merging the fine-grained senses (Chugur et al., 2002) (Navigli, 2006). This approach has two problems. First, it requires a stopping criterion for each word — for example the number of final classes. The right number of classes for each word cannot usually be predetermined even if the application is known. So such systems cannot be used to derive coarse senses for all the words. Second, inconsistent sense clusters are obtained because coarse senses are independently generated for each word. This leads to transitive closure errors and suggests that for deriving consistent coarse senses, instead of clustering senses for each word separately we should cluster synsets.

We propose a framework that derives a coarse sense inventory by learning a synset similarity metric. We focus on coarsening the noun synsets of WordNet and show that the obtained coarse-grained sense inventory greatly improves the noun sense disambiguation. Our approach closely resembles (Snow et al., 2007) for supervised learning of synset similarity. But to learn similarity between synset pairs which do not share a word we use a variant of the SimRank framework (Jeh and Widom, 2002) and avoid giving them zero similarity. Thus the similarity learnt is more than a binary decision and is reflective of a more comprehensive semantic similarity between the synsets. The use of SimRank for learning synset similarity is inspired by the success of graph-centrality algorithms in WSD. We do not modify the WordNet ontology, unlike (Snow et al., 2007), as it may introduce spurious relations and remove some manually encoded information.

Proceedings of the TextGraphs-8 Workshop, pages 11–19,
Seattle, Washington, USA, 18 October 2013.

In section 2, we discuss past work in sense clustering. In section 3 and 4, we describe our framework of learning synset similarity using SimRank. In section 5, we discuss our methodology of producing coarse senses using the learnt similarity metric. Section 6 describes the experimental setup and evaluates the framework described. Section 7 contains conclusions and discusses the directions for future work.

2 Related Work

A wide variety of automatic methods have been proposed for coarsening fine-grained inventories. The earliest attempt on WordNet include (Mihalcea and Moldovan, 2001) which merged synsets on semantic principles like sharing a *pertainym*, *antonym* or *verb group*. We discuss some of the ideas which are related to our work. Though promising, many of these techniques are severely limited by the amount of available manually annotated data.

(Chugur et al., 2002) constructed sense similarity matrices using *translation equivalences* in four languages. With the advent of WordNets being developed in multiple languages[1] as well as multilingual ontologies like BabelNet (Navigli and Ponzetto, 2012), this seems a promising area.

(McCarthy, 2006) estimated sense similarities using a combination of word-to-word distributional similarity combined with the JCN WordNet based similarity measure (Jiang and Conrath, 1997). They introduce a more relaxed notion of sense relatedness which allows the user to control the granularity for the application in hand.

(Navigli, 2006) produced a fixed set sense clusters by mapping WordNet word senses to Oxford English Dictionary(OED) word senses exploiting similarities in glosses and semantic relationships in the sense inventories. It is expected that the different WordNet senses that are semantically close mapped to the same sense in the other ontology via an efficient mapping that is able to capture the semantic similarity between the concepts in both the ontologies. The drawback of this method is the generation of inconsistent sense clusters.

(Snow et al., 2007) presented a novel supervised approach in which they train a Support Vector Machine(SVM) using features derived from WordNet and other lexical resources, whose predictions serve as a distance measure between synsets. Assuming zero similarity between synset pairs with no common words, they cluster synsets using average link agglomerative clustering and the synset similarity model learnt.

3 SimRank

SimRank (Jeh and Widom, 2002) is a graph based similarity measure applicable in any domain with object-to-object relationships. It uses the intuition that "*two objects are similar if they are related to similar objects*". Since SimRank has a recursive structure, the base cases play an important role.

Let us denote the SimRank similarity between objects α and β by $s(\alpha, \beta)$. It is defined as 1 if $\alpha = \beta$, otherwise it is given by:

$$s(\alpha,\beta) = \frac{C}{|I(\alpha)||I(\beta)|} \sum_{i=1}^{|I(\alpha)|} \sum_{j=1}^{|I(\beta)|} s(I_i(\alpha), I_j(\beta)) \quad (1)$$

where $C \in (0,1)$ is a constant decay factor and $I(v)$ is the set consisting of in-neighbours of node v, whose individual members are referred to as $I_j(v)$, $1 \leq j \leq |I(v)|$.

3.1 Solution and its Properties

(Jeh and Widom, 2002) proved that a solution $s(*,*)$ to the SimRank equations always exists and is unique. For a graph $G(V,E)$, the solution is reached by iteration to a fixed-point. For each iteration k, we keep $|V|^2$ entries $S_k(*,*)$, where $S_k(\alpha,\beta)$ is the estimate of similarity between α and β at the k^{th} iteration. We start with $S_0(*,*)$ which is 1 for singleton nodes like (x,x), 0 otherwise. We successively compute $S_{k+1}(*,*)$ based on $S_k(*,*)$ using equation 1.

Regarding the convergence of the above computation process, (Lizorkin et al., 2010) proved that the difference between the SimRank theoretical scores

[1] GlobalWordNet lists the WordNets available in the public domains: http://www.globalwordnet.org/gwa/wordnet_table.html.

and iterative similarity scores decreases exponentially in the number of iterations and uniformly for every pair of nodes i.e.

$$s(\alpha, \beta) - S_k(\alpha, \beta) \leq C^{k+1} \quad \forall \alpha, \beta \in V;\ k = 0, 1, 2 \ldots \tag{2}$$

3.2 Personalizing SimRank

In many scenarios we do not have complete information about the objects and thus have similarities for only some pairs of objects. These similarities may be independently learnt and may not directly conform with the underlying graph. In such situations, we would like to get a more complete and consistent similarity metric between objects while simultaneously using the existing information. For this we propose a personalized framework for SimRank where we bias the SimRank by changing the initialization. If we know similarities of some pairs, we fix them in our set of equations and let the rest of the values be automatically learnt by the system.

Let us call the map of node pairs to their similarity values as $InitStore$. It also contains all the singleton nodes like (x, x) which have values equal to 1. For other node pairs, the system of equations is the same as equation 1. In the personalized framework, we have no constraints on the initialization as long as all values initialized are in the range $[0, C]$.

3.3 Learning Synset Similarity using SimRank

The Personalized SimRank framework requires an underlying graph $G(V, E)$, where V is the set of objects to be clustered and E is the set of semantic links connecting these objects and an $InitStore$ containing the similarity values over some pairs from $V \times V$ learnt or known otherwise. Note that the values in the $InitStore$ have an upper bound of C.

For learning synset similarity, V is the set of synsets to be clustered and E is the set of WordNet relations connecting these synsets. We use the *Hypernymy*, *Hyponymy*, *Meronymy* and *Holonymy* relations of WordNet as the semantic links. The method for seeding the $InitStore$ is described in section 4 and can be summed up as follows:

- We train the SVMs from synset-merging data from OntoNotes (Hovy et al., 2006) to predict the similarity values of all the synset pairs which share at least one word.
- We estimate the posterior probabilities from the SVM predictions by approximating the posterior by a sigmoid function, using the method discussed in (Lin et al., 2003).
- We scale the posterior probabilities obtained to range between $[0, C]$ by linear scaling, where C is the SimRank decay parameter.

4 Seeding SimRank with supervision

4.1 Outline

We learn semantic similarity between different senses of a word using supervision, which allows us to intelligently combine and weigh the different features and thus give us an insight into how humans relate word senses. We obtain pairs of synsets which human-annotators have labeled as "merged" or "not merged" and describe each pair as a feature vector. We learn a synset similarity measure by using an SVM on this extracted dataset, where positive examples are the pairs which were merged and negative examples are the ones which were not merged by the annotators. We then calculate the posterior probability using the classifier score which is used as an estimate of the similarity between synsets constituting the pair.

4.2 Gold standard sense clustering dataset

Since our methodology depends upon the availability of labelled judgements of synset relatedness, a dataset with a high Inter-Annotator agreement is required. We use the manually labelled mappings from the Omega ontology[2] (Philpot et al., 2005) to the WordNet senses, provided by the OntoNotes project (Hovy et al., 2006).

The OntoNotes dataset creation involved a rigorous iterative annotation process producing a coarse sense inventory which guarantees at least 90% Inter-Tagger agreement on the sense-tagging of the sample sentences used in the annotation process. Thus we expect the quality of the final clustering of senses and the derived labelled judgements to be reasonably high.

[2] http://omega.isi.edu/

We use OntoNotes Release 3.0 [3] for extracting WordNet sense clusters.[4]. The dataset consists of senses for selected words in sense files. The senses in OntoNotes are mapped to WordNet senses, if a good mapping between senses exists. The steps involved in extraction are as follows:

1. OntoNotes has mappings to 4 WordNet versions: 1.7, 2.0, 2.1 and 3.0. We mapped all the senses[5] to WordNet 3.0.

2. Validating clusters on WN3.0:
   - We removed the sense files which did not contain all the senses of the word i.e. the clustering was not complete.
   - We removed the sense files in which the clusters had a clash i.e. one sense belonged to multiple clusters.

3. We removed instances that were present in both positive and negative examples. This situation arises because the annotators were working with word senses and there were inconsistent sense clusters.

Statistics	Nouns	Verbs
# of Word Sense File Before Processing	2033	2156
# of Word Sense Files After Processing	1680	1951
Distinct Offsets encountered	4930	6296
Positive Examples	1214	6881
Negative Examples	11974	20899
Percentage of Positive examples	9.20	24.76

Table 1: Statistics of Pairwise Classification Dataset obtained from OntoNotes

4.3 Feature Engineering

In this section, we describe the feature space construction. We derive features from the structure of WordNet and other available lexical resources. Our features can be broadly categorized into two parts: derived from WordNet and derived from other corpora. Many of the listed features are motivated by (Snow et al., 2007) and (Mihalcea and Moldovan, 2001).

[3] http://www.ldc.upenn.edu/Catalog/docs/LDC2009T24/OntoNotes-Release-3.0.pdf

[4] The OntoNotes groupings will be available through the LDC at http://www.ldc.upenn.edu

[5] We dropped WN1.7 as there were very few senses and the mapping from WN1.7 to WN3.0 was not easily available.

4.3.1 Features derived from WordNet

WordNet based features are further subdivided into similarity measures and features. Among the WordNet similarity measures, we used Path Based Similarity Measures: WUP (Wu and Palmer, 1994), LCH (Leacock et al., 1998); Information Content Based Measures: RES (Resnik, 1995), JCN (Jiang and Conrath, 1997), LIN (Lin, 1998); Gloss Based Heuristics (variants of Lesk (Lesk, 1986)): Adapted Lesk (Banerjee and Pedersen, 2002), Adapted Lesk Tanimoto and Adapted Lesk Tanimoto without hyponyms[6]

Other synset and sense based features include number of lemmas common in two synsets, SenseCount: maximum polysemy degree among the lemmas shared by the synsets, SenseNum: number of lemmas having maximum polysemy degree among the lemmas shared by the synsets, whether two synsets have the same lexicographer file, number of common hypernyms, autohyponymy: whether the two synsets have a hyponym-hypernym relation between them and merging heuristics by (Mihalcea and Moldovan, 2001).[7]

4.3.2 Features derived from External Corpora

- eXtended WordNet Domains Project (González et al., 2012) provides us the score of a synset with respect to 169 hierarchically organized domain-labels(excluding factotum label). We obtain a representation of a synset in the domain label space and use cosine similarity, L1 distance and L2 distance computed over the weight representations of the synsets as features.

- BabelNet (Navigli and Ponzetto, 2012) provides us with the translation of noun word senses in 6 languages namely: English, German, Spanish, Catalan, Italian and French and the mapping of noun synsets to DBpedia[8] entries. For features we use counts of common

[6] We call the lesk variants as AdapLesk, AdapLeskTani and AdapLeskTaniNoHypo.

[7] We divide mergeSP1_2 into two features: The strict heuristic checks whether all the hypernyms are shared or not whereas the relaxed heuristic checks if the synsets have at least 1 common hypernym.

[8] http://dbpedia.org/About

lemmas in all 6 languages and count of common DBpedia entries.

- SentiWordNet (Baccianella et al., 2010) provides us with a mapping from a synset to a triad of three weights. The weights correspond to the score given to a synset based on its objectivity and subjectivity(positive and negative). We use cosine similarity, L1 distance and L2 distance of the weight representations of the synsets as features.

- We use the sense clusterings produced by mapping WordNet senses to OED senses by the organizers of the coarse-grained AW task in SemEval-2007[9] (Navigli et al., 2007). For each pair of synsets, we check if there are senses in the synsets that belong to the same cluster in the OED mapping.

4.4 Classifier and Training

We train SVMs using the features above on the synset pairs extracted from OntoNotes, where every synset pair is given either a "merged" or "not-merged" label. Because of the skewed class distribution in the dataset, we randomly generated balanced datasets (equal number of positive and negative instances) and then divided them in a ratio of 7:3 for training and testing respectively. We repeated the process multiple number of times and report the average.

To train the SVMs we used an implementation by (Joachims, 1998), whose java access is provided by JNI-SVMLight [10] library. For all experiments reported, we use the linear kernel with the default parameters provided by the library. [11]

We scale the ranges of all the features to a common range [-1,1]. The main advantage offered by scaling is that it prevents domination of attributes with smaller numeric ranges by those with greater numeric ranges. It also avoids numerical difficulties like overflow errors caused by large attribute values. Note that both training and testing data should be scaled with the same parameters.

[9] http://lcl.uniroma1.it/coarse-grained-aw/

[10] JNI-SVMLight: http://adrem.ua.ac.be/~tmartin/

[11] We also tested our system with an RBF kernel but the best results were obtained with the linear kernel(Bhagwani, 2013)

4.5 Estimating Posterior Probabilities from SVM Scores

For seeding SimRank, we need an estimate of the posterior probability $Pr(y = +1|x)$ instead of the class label. (Platt, 1999) proposed approximating the posterior by a sigmoid function

$$Pr(y = +1|x) \approx P_{A,B}(f(x)) \equiv \frac{1}{1+exp(Af(x)+B)}$$

We use the method described in (Lin et al., 2003), as it avoids numerical difficulties faced by (Platt, 1999).

5 Coarsening WordNet

We construct an undirected graph $G(V, E)$ where the vertex set V contains the synsets of WordNet and edge set E comprises of edges obtained by thresholding the similarity metric learnt using the personalized SimRank model (see section 3.2). On varying the threshold, we obtain different graphs which differ in the number of edges. On these graphs, we find connected components[12], which gives us a partition over synsets. All the senses of a word occurring in the same component are grouped as a single coarse sense. We call our approach Connected Components Clustering(CCC).

For lower thresholds, we obtain denser graphs and thus fewer connected components. This small number of components translates into more coarser senses. Therefore, using this threshold as a parameter of the system, we can control the granularity of the coarse senses produced.

6 Experimental Setup and Evaluation

6.1 Feature Analysis

We analyze the feature space used for SVMs in two ways. We evaluate Information Gain(IG) and Gain Ratio(GR) functions over the features and do a feature ablation study. The former tries to capture the discrimination ability of the feature on its own and the latter measures how a feature corroborates with other features in the feature space.

[12] a connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph.

We extracted all the features over the complete OntoNotes dataset without any normalization and evaluated them using IG and GR functions. We report the top 7 features of both the evaluators in table 2[13].

Feature	GR	IG
LCH	0.0129	**0.0323**
WUP	0.0148	**0.0290**
JCN	**0.0215**	0.0209
AdapLesk	0.0169	**0.0346**
AdapLeskTani	**0.0231**	**0.0360**
AdapLeskTaniNoHypo	0.0168	**0.0301**
mergeSP1_2_strict	**0.0420**	0.0010
mergeSP1_2_relaxed	**0.0471**	0.0012
number of Common Hypernyms	**0.0883**	0.0096
Domain-Cosine Similarity	**0.0200**	**0.0442**
OED	**0.0326**	**0.0312**

Table 2: Information Gain and Gain Ratio Based Evaluation

We divide our features into 6 broad categories and report the average F-Score of both the classes observed by removing that category of features from our feature space. The SVMs are trained with features normalized using MinMax Normalization for this study.

Features Removed	FScore Pos	FScore Neg
WordNet Similarity Measures	0.6948	0.6784
WordNet Based Features	0.7227	0.7092
BabelNet Features	0.7232	0.7127
Domain Similarity Features	0.6814	0.6619
OED Feature	0.6957	0.7212
SentiWordNet Features	0.7262	0.7192
Without Removing Features	0.7262	0.7192

Table 3: Feature Ablation Study

From tables 2 and 3, we observe that the most significant contributors in SVM performance are WordNet similarity measures and domain cosine similarity. The former highlights the importance of the ontology structure and the gloss definitions in WordNet. The latter stresses the fact that approximately matching the domain of two senses is a strong cue about whether the two senses are semantically related enough to be merged.

Other notable observations are the effectiveness of the OED feature and the low Information Gain and Gain Ratio of multilingual features. We also found that SentiWordNet features were non-discriminatory as most of the noun synsets were described as objective concepts.

6.2 Estimating Posterior Probabilities from SVM Scores

We learn parameters A and B of the sigmoid that transforms SVM predictions to posterior probabilities (see section 4.5). Since using the same data set that was used to train the model we want to calibrate will introduce unwanted bias we calibrate on an independently generated random balanced subset from OntoNotes.

The values of A and B obtained are -1.1655 and 0.0222 respectively. Using these values, the SVM prediction of value 0 gets mapped to 0.4944.

6.3 Semi-Supervised Similarity Learning

We learn similarity models using the SimRank variant described in section 3. (Jeh and Widom, 2002) use $C = 0.8$ and find that 5-6 iterations are enough. (Lizorkin et al., 2010) suggest lower values of C or more number of iterations. We vary the values for C between 0.6, 0.7 and 0.8 and we run all systems for 10 iterations to avoid convergence issues.

6.4 Coarsening WordNet

We assess the effect of automatic synset clustering on the English all-words task at Senseval-3 (Snyder and Palmer, 2004) [14]. The task asked WSD systems to select the apt sense for 2,041 content words in running texts comprising of 351 sentences. Since the BabelNet project provided multilingual equivalences for only nouns, we focussed on nouns and used the 890 noun instances.

We consider the three best performing WSD systems: GAMBL (Decadt et al., 2004), SenseLearner (Mihalcea and Faruque, 2004) and Koc University (Yuret, 2004) - and the best unsupervised system: IRST-DDD (Strapparava et al., 2004) submitted in the task. The answer by the system is given full

[13]Table lists only 11 features as 3 features are common in top 7 features of both the evaluators

[14]This evaluation is similar to the evaluation used by (Navigli, 2006) and (Snow et al., 2007)

C	System	F-Score	Threshold	CCC	Random	Improvement
0.6	GAMBL	0.7116	0.36	0.9031	0.8424	0.0607
	SenseLearner	0.7104	0.37	0.8824	0.8305	0.0518
	KOC University	0.7191	0.37	0.8924	0.8314	0.0610
	IRST-DDD	0.6367	0.35	0.8731	0.8013	0.0718
0.7	GAMBL	0.7116	0.52	0.8453	0.7864	0.0589
	SenseLearner	0.7104	0.49	0.8541	0.8097	0.0444
	KOC University	0.7191	0.52	0.8448	0.7911	0.0538
	IRST-DDD	0.6367	0.49	0.7970	0.7402	0.0568
0.8	GAMBL	0.7116	0.59	0.8419	0.7843	0.0577
	SenseLearner	0.7104	0.56	0.8439	0.7984	0.0455
	KOC University	0.7191	0.59	0.8414	0.7879	0.0535
	IRST-DDD	0.6367	0.47	0.8881	0.8324	0.0557

Table 4: Improvement in Senseval-3 WSD performance using Connected Component Clustering Vs Random Clustering at the same granularity

credit if it belongs to the cluster of the correct answer.

Observe that any clustering will only improve the WSD performance. Therefore to assess the improvement obtained because of our clustering, we calculate the expected F-Score, the harmonic mean of expected precision and expected recall, for a random clustering at the same granularity and study the improvement over the random clustering.

Let the word to be disambiguated have N senses, each mapped to a unique synset. Let the clustering of these N synsets on a particular granularity give us k clusters $C_1, \ldots C_k$. The expectation that an incorrectly chosen sense and the actual correct sense would belong to same cluster is

$$\frac{\sum_{i=1}^{k}|C_i|(|C_i|-1)}{N(N-1)} \quad (3)$$

We experiment with C = 0.6, 0.7 and 0.8. The SVM probability boundaries when scaled to $[0, C]$ for these values are 0.30, 0.35 and 0.40. To find the threshold giving the best improvement against the random clustering baseline, we use the search space $[C - 0.35, C]$. The performance of the systems at these thresholds for different values of C is reported in table 4.

Commenting theoretically about the impact of C on the performance is tough as by changing C we are changing all the $|V|^2$ simultaneous equations to be solved. Empirically, we observe that across all systems improvements over the baseline keep decreasing as C increases. This might be due to the slow convergence of SimRank for higher values of C.

Figure 1 shows that by varying thresholds the improvement of the Connected Components Clustering over the random clustering baseline at the same granularity first increases and then decreases. This behaviour is shared by both supervised and unsupervised systems. Similar figures are obtained for other values of C (0.7 and 0.8), but are omitted because of lack of space.

Across supervised and unsupervised systems, we observe higher improvements for unsupervised systems. This could be because the unsupervised system was underperforming compared to the supervised systems in the fine grained WSD task setting.

7 Conclusions and Future Work

We presented a model for learning synset similarity utilizing the taxonomy information and information learnt from manually obtained sense clustering. The framework obtained is generic and can be applied to other parts of speech as well. For coarsening senses, we used one of the simplest approaches to cluster senses but the generic nature of the similarity gives us the flexibility to use other clustering algorithms

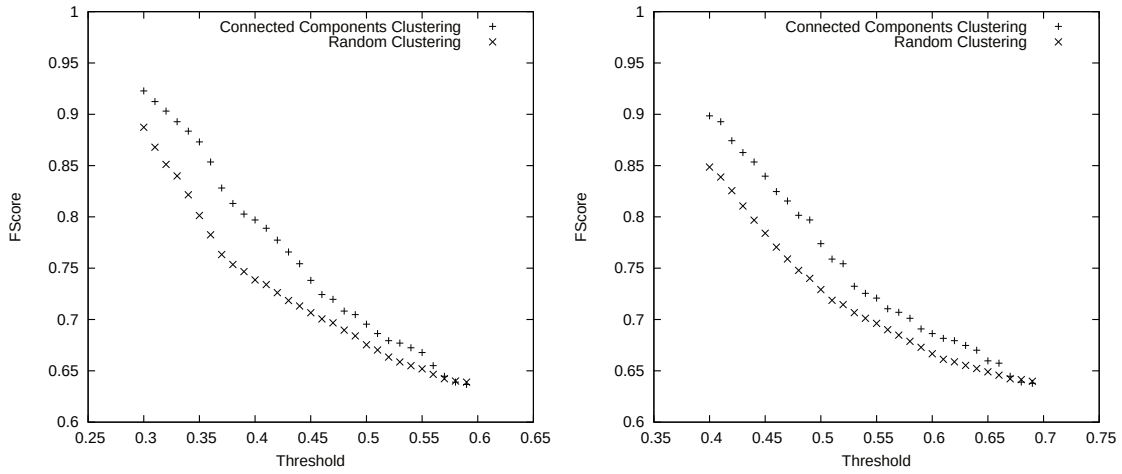

Figure 1: Improvement in (a) average performance of best 3 Supervised Systems and (b) performance of best Unupervised System in Senseval-3 using Connected Component Clustering Vs Random Clustering at the same granularity with C = 0.6

for experimentation. We show that the clustering obtained by partitioning synsets in connected components gives us a maximum improvement of 5.78% on supervised systems and 7.18% on an unsupervised system. This encourages us to study graph based similarity learning methods further as they allow us to employ available wide-coverage knowledge bases.

We use the WordNet relations *Hypernymy*, *Hyponymy*, *Meronymy* and *Holonymy* without any differentiation. If we can grade the weights of the relations based on their relative importance we can expect an improvement in the system. These weights can be obtained by annotator feedback from cognitive experiments or in a task based setting. In addition to the basic WordNet relations, we can also enrich our relation set using the Princeton WordNet Gloss Corpus[15], in which all the WordNet glosses have been sense disambiguated. Any synset occuring in the gloss of a synset is directly related to that synset via the *gloss* relation. This relation helps make the WordNet graph denser and richer by capturing the notion of *semantic relatedness*, rather than just the notion of *semantic similarity* captured by the basic WordNet relations.

[15]http://wordnet.princeton.edu/glosstag.shtml

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of LREC*.

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of CICLing 2002*.

Sumit Bhagwani. 2013. Merging word senses. Master's thesis, Indian Institute of Technology Kanpur.

Paul Buitelaar. 2000. Reducing lexical semantic complexity with systematic polysemous classes and underspecification. In *NAACL-ANLP 2000 Workshop: Syntactic and Semantic Complexity in Natural Language Processing Systems*, pages 14–19. Association for Computational Linguistics.

Irina Chugur, Julio Gonzalo, and Felisa Verdejo. 2002. Polysemy and sense proximity in the senseval-2 test suite. In *Proceedings of the ACL 2002 WSD workshop*.

Bart Decadt, Véronique Hoste, Walter Daelemans, and Antal Van den Bosch. 2004. Gambl, genetic algorithm optimization of memory-based wsd. In *Proceedings of ACL/SIGLEX Senseval-3*.

Aitor González, German Rigau, and Mauro Castillo. 2012. A graph-based method to improve wordnet domains. In *Proceedings of CICLing 2012*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of HLT-NAACL 2006*.

Glen Jeh and Jennifer Widom. 2002. Simrank: A measure of structural-context similarity. In *KDD*, pages 538–543.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of ROCLING'97*.

Thorsten Joachims. 1998. Making large-scale support vector machine learning practical.

Claudia Leacock, George A. Miller, and Martin Chodorow. 1998. Using corpus statistics and wordnet relations for sense identification. *Comput. Linguist.*, 24(1):147–165, March.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC 1986*.

Hsuan-tien Lin, Chih-Jen Lin, and Ruby C. Weng. 2003. A note on platt's probabilistic outputs for support vector machines.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of ICML 1998*.

Dmitry Lizorkin, Pavel Velikhov, Maxim Grinev, and Denis Turdakov. 2010. Accuracy estimate and optimization techniques for simrank computation. *The VLDB Journal*, 19(1):45–66, February.

Diana McCarthy. 2006. Relating wordnet senses for word sense disambiguation. *Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*, 17.

Rada Mihalcea and Ehsanul Faruque. 2004. Senselearner: Minimally supervised word sense disambiguation for all words in open text. In *Proceedings of ACL/SIGLEX Senseval-3*.

Rada Mihalcea and Dan Moldovan. 2001. Ez.wordnet: principles for automatic generation of a coarse grained wordnet. In *Proceedings of Flairs 2001*, pages 454–459.

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

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of SemEval-2007*, pages 30–35. Association for Computational Linguistics, June.

Roberto Navigli. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of COLING-ACL*, pages 105–112.

Hwee Tou Ng, Bin Wang, and Yee Seng Chan. 2003. Exploiting parallel texts for word sense disambiguation: an empirical study. In *Proceedings of ACL 2003*.

Andrew Philpot, Eduard Hovy, and Patrick Pantel. 2005. The omega ontology. In *Proceedings of the ONTOLEX Workshop at IJCNLP 2005*.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI 1995*.

Rion Snow, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng. 2007. Learning to Merge Word Senses. In *Proceedings of EMNLP-CoNLL*, pages 1005–1014, June.

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In *Proceedings of ACL/SIGLEX Senseval-3*, pages 41–43.

Carlo Strapparava, Alfio Gliozzo, and Claudiu Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation: Irst at senseval-3. In *Proceedings of ACL/SIGLEX Senseval-3*.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of ACL 1994*.

Deniz Yuret. 2004. Some experiments with a naive bayes wsd system. In *Proceedings of ACL/SIGLEX Senseval-3*.