Taxonomy Learning Using Word Sense Induction

Ioannis P. Klapaftis
Department of Computer Science
The University of York
York, UK, YO10 5DD
gianni s@cs.york.ac.uk

Suresh Manandhar
Department of Computer Science
The University of York
York, UK, YO10 5DD
suresh@cs.york.ac.uk

Abstract

Taxonomies are an important resource for a variety of Natural Language Processing (NLP) applications. Despite this, the current state-of-the-art methods in taxonomy learning have disregarded word polysemy, in effect, developing taxonomies that conflate word senses. In this paper, we present an unsupervised method that builds a taxonomy of senses learned automatically from an unlabelled corpus. Our evaluation on two WordNet-derived taxonomies shows that the learned taxonomies capture a higher number of correct taxonomic relations compared to those produced by traditional distributional similarity approaches that merge senses by grouping the features of each word into a single vector.

1 Introduction

A concept or a sense, s, can be defined as the meaning of a word or a multiword expression. A concept s can be linguistically realised by more than one word while at the same time a word w can be the linguistic realisation of more than one concept. Given a set of concepts S, taxonomy learning is the task of hierarchically classifying the elements in S in an *automatic* manner. For example, consider a set of concepts linguistically realised by the words/multiword expressions *LAN, computer network, internet, meshwork, gauze, snood*. Taxonomy learning methods produce taxonomies, such as the ones shown in Figures 1 (a) and 1 (b).

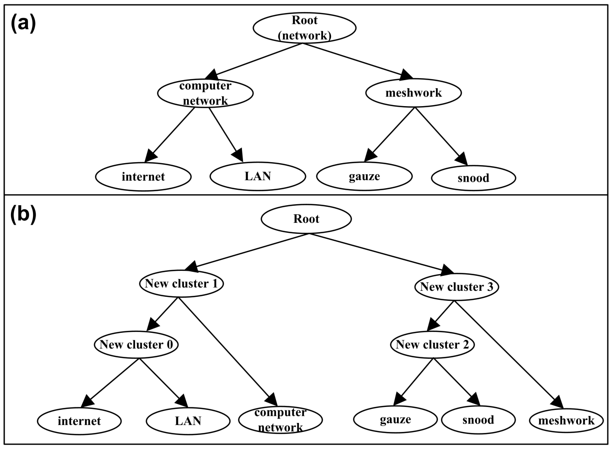


Figure 1: A labelled and an unlabelled concept taxonomy

By observing Figure 1 (a), we can express **IS-A** statements, such as *Internet* **IS-A** *Computer Network* etc. However, the same does not apply to the taxonomy in Figure 1 (b), since this taxonomy is not fully labelled. Despite this, its hierarchical organisation clearly shows that the concepts are divided into groups, which are further subdivided into subgroups and so forth, until we reach a level where each concept belongs to its own group. Unlabelled taxonomies are typically produced by agglomerative hierarchical clustering algorithms (King, 1967; Sneath and Sokal, 1973).

The knowledge encoded in taxonomies can be utilised in a range of NLP applications. For instance, taxonomies can be used in information retrieval to expand a user query with semantically related words or to enhance document representation by abstracting from plain words and adding conceptual information (Cimiano, 2006). WordNet's (Fellbaum, 1998) taxonomic relations have also been used in Word Sense Disambiguation (WSD) (Navigli and Velardi, 2004b). In named entity recognition, methods relying on gazetteers could make

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 82–90,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

use of automatically acquired taxonomies (Cimiano, 2006), while question answering systems have also benefited (Moldovan and Novischi, 2002).

Despite the wide uses of taxonomies, the majority of methods disregard or do not deal effectively with word polysemy, in effect, developing taxonomies that conflate the senses of words (see Section 2). In this work, we show that Word Sense Induction (WSI) can be effectively employed to address this limitation of existing methods.

We present a novel method that employs WSI to generate the different senses of a set of target words from an unlabelled corpus and then produces a taxonomy of senses using Hierarchical Agglomerative Clustering (HAC) (King, 1967; Sneath and Sokal, 1973). We evaluate our method on two WordNet-derived sub-taxonomies and show that our method leads to the development of concept hierarchies that capture a higher number of correct taxonomic relations in comparison to those generated by current distributional similarity approaches.

2 Related work

Initial research on taxonomy learning focused on identifying in a given text lexico-syntactic patterns that suggest hyponymy relations (Hearst, 1992). For instance, the pattern NP_0 *such as* NP_1,…,NP_n suggests that NP_0 is a hypernym of NP_i. For example, given the phrase *Fruits, such as oranges, apples,...*, the above pattern would suggest that *fruit* is a hypernym of *orange* and *apple*. These pattern-based approaches operate at the word level by learning lexical relations between words rather than between senses of words.

In the same spirit, other work attempted to exploit the regularities of dictionary entries to identify hyponymy relations (Amsler, 1981). For example in WordNet, *WAN* is defined as *a computer network that spans* …. Hence, one can easily induce that *WAN* is a hyponym of *computer network* by assuming that the first noun phrase in the definition is a hypernym of the target word. These approaches learn lexical relations at the sense level since dictionaries separate the senses of a word. However this would be true if and only if the glosses of the dictionaries were sense-annotated, which is not the case for the majority of electronic dictionaries (Cimiano, 2006). Another limitation is that taxonomies are built according to the sense distinctions present in dictionaries and not according to the actual use of words in the corpus.

The majority of taxonomy learning approaches are based on the *distributional hypothesis* (Harris, 1968). Typically, distributional similarity methods (Cimiano et al., 2004; Cimiano et al., 2005; Faure and Nédellec, 1998; Reinberger and Spyns, 2004; Caraballo, 1999) utilise syntactic dependencies such as subject/verb, object/verb relations, conjunctive and appositive constructions and others. These dependencies are used to extract the features that serve as the dimensions of the vector space. Each target noun is then represented as a vector of extracted features where the frequency of co-occurrence of the target noun with each feature is used to calculate the weight of that feature. The constructed vectors are the input to hierarchical clustering or *formal concept analysis* (Ganter and Wille, 1999) to produce a taxonomy. These approaches assume that a target noun is monosemous creating one vector of features for each target noun. This limitation can lead to a number of problems.

Firstly, the constructed taxonomies might be biased towards the inclusion of taxonomic relationships between the most frequent senses of target nouns, ignoring interesting taxonomic relations where less frequent senses are present. For example, consider the word *house*. Current distributional similarity methods would possibly capture the hyponyms of its Most Frequent Sense (MFS[1]), however ignoring the hyponyms of less frequent senses of *house*, e.g. *casino*, *theater*, etc. Given that word senses typically follow a Zipf distribution, these methods construct vectors dominated by the MFS of words. This bias significantly degrades the usefulness of learned taxonomies.

Secondly, given that distributional similarity approaches rely on the computation of pairwise similarities between target words, merging their senses to a single vector might lead to unreliable similarity estimates. For example, merging the features of the different senses of *house* could provide a lower similarity with its monosemous hyponym *beach house*, since only the first sense of *house* is related to *beach*

[1] WordNet: A dwelling that serves as living quarters …

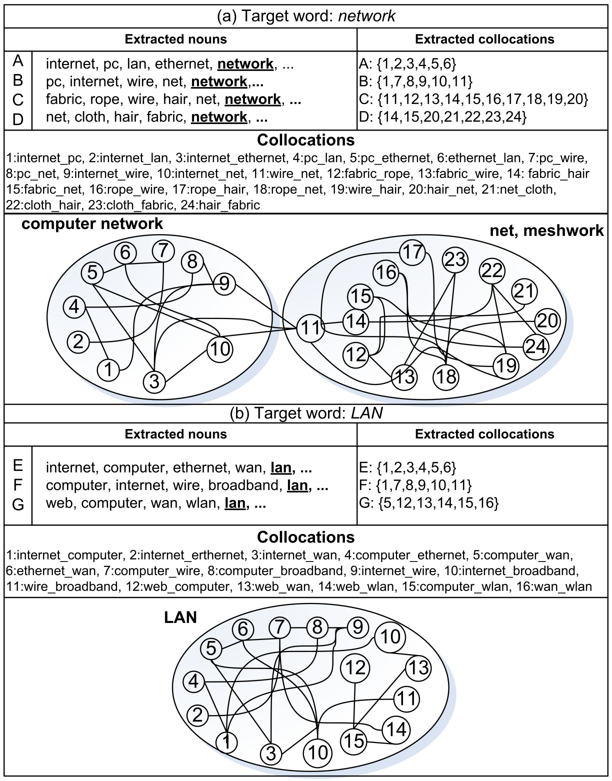
house. This problem might lead both to inclusion of incorrect or loss of correct taxonomic relations. In our work, we aim to overcome these drawbacks by identifying the different senses with which target words appear in text and then building a hierarchy of the identified senses.

Soft clustering approaches (Reinberger and Spyns, 2004; Reinberger et al., 2003) have also been applied to taxonomy learning to deal with polysemy. These methods associate each verb with a vector of features, where each feature is a noun appearing as a subject or object of that verb. That way a noun can appear in different vectors, hence in different clusters during hierarchical clustering as a result of its polysemy. However, the underlying assumption is that a verb is monosemous with respect to its associated vector of nouns. This assumption is not always valid and can cause the problems mentioned above.

Other work in taxonomy learning exploits the head/modifier relationships to create taxonomic relations (Buitelaar et al., 2004; Hwang, 1999; Sánchez and Moreno, 2005). These relations are used to create: (1) a class (concept) for each head, and (2) subclasses by adding nominal or adjectival modifiers. For example, *credit card* **IS-A** *card*. The corresponding hyponymy relations are learned at the lexical level disregarding word polysemy. Some of these approaches identified the problem of polysemy and applied sense disambiguation with respect to WordNet in order to capture the different senses of a target term (Navigli and Velardi, 2004b; Navigli and Velardi, 2004a). Specifically, the taxonomy built by exploiting head/modifiers relations was modified according to WordNet's hyponymy relations between senses of disambiguated terms. One important deficiency of using sense disambiguation is that dictionaries miss many domain-specific senses. Additionally, the fixed-list of senses paradigm prohibits learning word senses according to their use in context. The use of sense induction we propose in this paper aims to overcome these limitations.

3 Method

Given a set of words W, a WSI method is applied to each $w_i \in W$ (Section 3.1). The outcome of the first stage is a set of senses, S, where each $s_i^w \in S$ denotes the i-th sense of word $w \in W$. This set of senses is the input to hierarchical clustering that produces a hierarchy of senses (Section 3.2).



(a) Target word: *network*

	Extracted nouns	Extracted collocations
A	internet, pc, lan, ethernet, **network**, ...	A: {1,2,3,4,5,6}
B	pc, internet, wire, net, **network**,...	B: {1,7,8,9,10,11}
C	fabric, rope, wire, hair, net, **network**, ...	C: {11,12,13,14,15,16,17,18,19,20}
D	net, cloth, hair, fabric, **network**, ...	D: {14,15,20,21,22,23,24}

Collocations

1:internet_pc, 2:internet_lan, 3:internet_ethernet, 4:pc_lan, 5:pc_ethernet, 6:ethernet_lan, 7:pc_wire, 8:pc_net, 9:internet_wire, 10:internet_net, 11:wire_net, 12:fabric_rope, 13:fabric_wire, 14: fabric_hair 15:fabric_net, 16:rope_wire, 17:rope_hair, 18:rope_net, 19:wire_hair, 20:hair_net, 21:net_cloth, 22:cloth_hair, 23:cloth_fabric, 24:hair_fabric

(b) Target word: *LAN*

	Extracted nouns	Extracted collocations
E	internet, computer, ethernet, wan, **lan, ...**	E: {1,2,3,4,5,6}
F	computer, internet, wire, broadband, **lan, ...**	F: {1,7,8,9,10,11}
G	web, computer, wan, wlan, **lan, ...**	G: {5,12,13,14,15,16}

Collocations

1:internet_computer, 2:internet_erthernet, 3:internet_wan, 4:computer_ethernet, 5:computer_wan, 6:ethernet_wan, 7:computer_wire, 8:computer_broadband, 9:internet_wire, 10:internet_broadband, 11:wire_broadband, 12:web_computer, 13:web_wan, 14:web_wlan, 15:computer_wlan, 16:wan_wlan

Figure 2: WSI for *network* & *LAN*

3.1 Word sense induction

WSI is the task of identifying the senses of a target word in a given text. Recent WSI methods were evaluated under the framework of SemEval-2007 WSI task (SWSI) (Agirre and Soroa, 2007). The evaluation framework defines two types of assessment, i.e. evaluation in: (1) a clustering and (2) a WSD setting. Based on this evaluation, we selected the method of Klapaftis & Manandhar (2008) (henceforth referred to as KM) that achieves high F-score in both evaluation schemes as compared to the systems participating in SWSI. We briefly describe KM mentioning its parameters used in our evaluation (Section 4). Figures 2 (a) and 2 (b) describe the different steps for inducing the senses of the target words *network* and *LAN*.

Corpus preprocessing: The input to KM is a base corpus *bc*, in which the target word w appears in each paragraph. In Figure 2 (a), the base corpus consists of the paragraphs *A*, *B*, *C* and *D*. The aim of this stage is to capture nouns contextually related to w. Initially, the target word is removed from bc, part-of-speech tagging is applied to each paragraph, only nouns are kept and lemmatised. In the next step, the distribution of each noun is compared to the distribution of the same noun in a reference corpus[2] using the log-likelihood ratio (G^2) (Dunning, 1993). Nouns with a G^2 below a pre-specified threshold (parameter p_1) are removed from each paragraph. Figure 2 (a) shows the remaining nouns for each paragraph of bc.

Graph creation & clustering: In the setting of KM, a collocation is a juxtaposition of two nouns within the same paragraph. Thus, each noun is combined with any other noun yielding a total of $\binom{N}{2}$ collocations for a paragraph with N nouns. Each collocation, c_{ij}, is assigned a weight that measures the relative frequency of two nouns co-occurring. This weight is the average of the conditional probabilities $p(n_i|n_j)$ and $p(n_j|n_i)$, where $p(n_i|n_j) = \frac{f(c_{ij})}{f(n_j)}$, $f(c_{ij})$ is the number of paragraphs nouns n_i, n_j co-occur and $f(n_j)$ is the number of paragraphs in which n_j appears. Collocations are filtered with respect to their frequency (parameter p_2) and weight (parameter p_3). Each retained collocation is represented as a vertex. Edges between vertices are present, if two collocations co-occur in one or more paragraphs. Figure 2 (a) shows that this process has generated 24 collocations for the target word *network*. On the top right of the figure we also observe the collocations associated with each paragraph.

In the next step, a smoothing technique is applied to discover new edges between vertices. The weight applied to each edge connecting vertices v_i and v_j (collocations c_{ab}, c_{de}) is the maximum of their conditional probabilities ($\max(p(c_{ab}|c_{de}), p(c_{de}|c_{ab}))$). Finally, the graph is clustered using Chinese whispers (Biemann, 2006). The final output is a set of senses, each one represented by a set of contextually related collocations. In Figure 2, we generated two senses for *network* and one sense for *LAN*.

3.2 Hierarchical clustering of senses

Given the set of senses S, our task at this point is to hierarchically classify the senses using HAC. Consider for example the words *network* and *LAN*, and let us assume that the WSI process has generated the senses in Figures 2 (a) and 2 (b). HAC operates by treating each sense as a singleton cluster and then successively merging the most similar clusters according to a pre-defined similarity function. This process iterates until all clusters have been merged into a single cluster taken to be the *root*.

Senses	computer network	meshwork	LAN
computer network	1	0.0	0.66
meshwork	0.0	1	0.14
LAN	0.66	0.14	1

Table 1: Similarity matrix for HAC.

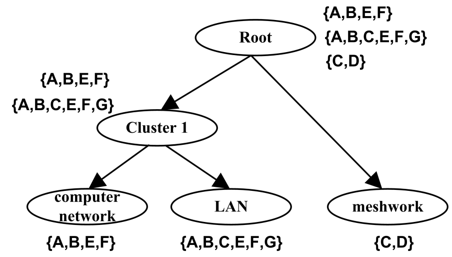


Figure 3: WSI & HAC example

To calculate the pairwise similarities between senses we exploit the attributes that represent each sense, i.e. their collocations. Let BC be the corpus resulting from the union of the base corpora of all words in W. In our example, BC would consist of the paragraphs, in which the words *network* and *LAN* appear, i.e. *A*, *B*, ..., *G*. An induced sense tags a paragraph, if one or more of its collocations appear in that paragraph. Thus, each induced sense is associated with a set of paragraph labels that denote the paragraphs tagged by that sense. Figure 3 shows the paragraph labels tagged by each sense of our example. Finally, given two senses s_i^a, s_i^b and their corresponding sets of tagged paragraphs f_i^a and f_i^b, we use the Jaccard coefficient to calculate their similarity, i.e. $\mathrm{JC}(s_i^a, s_i^b) = \frac{|\mathrm{f}_i^a \cap \mathrm{f}_i^b|}{|\mathrm{f}_i^a \cup \mathrm{f}_i^b|}$, where s_j^k denotes the j-th sense of word k. The resulting similarity matrix of our example is shown in Table 1. Given that matrix, HAC would first group *computer network* and *LAN* as they have the highest similarity (Figure 3). In the final iteration, the remaining two clusters (*Cluster 1* & *meshwork*) would be grouped to the *root*.

An important parameter of HAC is the choice of the technique for calculating cluster similarities. Note that as we move towards the higher levels of

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

the taxonomy clusters contain more than one sets of tagged paragraphs (Figure 3 - *Cluster 1*), hence the choice of the similarity function is crucial. We experiment with three techniques, i.e. *single-linkage*, *complete-linkage* and *average-linkage*. The first one defines the similarity between two clusters as the maximum similarity among all the pairs of their corresponding feature sets. The second considers the minimum similarity among all the pairs, while the third calculates the average similarity of all the pairs.

4 Evaluation

We evaluate our method with respect to two WordNet-derived sub-taxonomies (Section 4.3). For that reason, it is necessary to map the induced senses to WordNet before applying HAC. Note that the mapping process might map more than one induced senses to the same WordNet sense. In that case, these induced senses are merged to a single one along with their corresponding collocations.

4.1 Mapping WSI clusters to WordNet senses

The process of mapping the induced senses to WordNet is straightforward. Let $w \in W$ be a word with n senses in WordNet. A WordNet sense i of w is denoted by ws_i^w, $i = [1, n]$. Let us also assume that the WSI method has produced m senses for w, where each sense j is denoted as s_j^w, $j = [1, m]$. Each induced sense s_j^w is associated with a set of features f_j^w as in the previous section. These features are the paragraphs (paragraph labels) of BC tagged by s_j^w. In the next step, each WordNet sense ws_i^w is associated with its WordNet signature g_i^w that contains the following semantic features: hypernyms/hyponyms, meronyms/holonyms and synonyms of ws_i^w. For example, the signature of the fifth WordNet sense of *network* would contain *internet*, *cyberspace* and other semantically related words. Table 2 shows partial signatures for each sense of *network*.

The signature g_i^w is used to formalise the WordNet sense ws_i^w as a set of features q_i^w. These features are the paragraphs (paragraph labels) of BC that contain one or more of the aforementioned semantically related to ws_i^w words that exist in g_i^w. Given an induced sense s_j^w, a similarity score is calculated between s_j^w and each WordNet sense of w. The maximum score determines the WordNet sense label that will be assigned to s_j^w, i.e. $label(s_j^w) = \text{argmax}_i \, JC(f_j^w, q_i^w)$, where JC is the Jaccard similarity coefficient. In the example of Figure 2 (a), the *computer network* sense would be mapped to the fifth WordNet sense of *network*, since there is a significant overlap between the paragraphs tagged by the induced and that WordNet sense.

WordNet sense	Semantically related words/phrases
1	reticulum, RF, RAS
2	communication system/equipment
3	gauze, snood, tulle
4	reseau, reticle, reticulation
5	net, internet, cyberspace

Table 2: Semantically related words/phrases to *network*

4.2 Evaluation measures

For the purposes of this section we present one gold standard taxonomy (Figure 1 (a)) and a second derived from our method (Figure 1 (b)). The comparison of these taxonomies is based on the *semantic cotopy* of a node, which has also been used in (Maedche and Staab, 2002; Cimiano et al., 2005). In particular, the semantic cotopy of a node is defined as the set of all its super- and subnodes excluding the *root* and including that node. For example, the semantic cotopy of *computer network* in Figure 1 (a) is $\{$*computer network, internet, LAN*$\}$. There are two issues, which make the evaluation difficult.

The first one is that HAC produces a taxonomy in which all internal nodes are unlabelled, as opposed to the gold standard taxonomy. In Figure 1 (b), we have manually labelled internal nodes with their IDs for clarity. For example, the semantic cotopy of the node *New Cluster 1* in Figure 1 (b) is $\{$*computer network, internet, LAN, New Cluster 1, New Cluster 0*$\}$. By comparing the cotopies of nodes *computer network* in Figure 1 (a) and *New Cluster 1* in Figure 1 (b), we observe that the automatic method has successfully grouped all of the hypernyms and hyponyms of *computer network* under *New Cluster 1*. However, the corresponding cotopies are not identical, because the cotopy of *New Cluster 1* also includes the labels produced by HAC.

To deal with this problem, we use a version of semantic cotopy for nodes in the automatically learned taxonomy which excludes nodes that do not exist in WordNet. That way the semantic cotopies of *New Cluster 1* in Figure 1 (b) and *computer network* in

Figure 1 (a) will yield maximum similarity.

The second issue is that the nodes that exist in the gold standard taxonomy are leaf nodes in the automatically learned taxonomy. As a result, the semantic cotopy of *LAN* in Figure 1 (b) is $\{LAN\}$ since all of its supernodes do not exist in WordNet. In contrast, the semantic cotopy of *LAN* in Figure 1 (a) is {*LAN, computer network*}. We observe that there is an overlap between the two cotopies derived by the existence of the same concept in both taxonomies, i.e. *LAN*. In fact, all of the leaf nodes of a learned taxonomy will have a small overlap with the corresponding concept in the gold standard. For this problem, we observe that in our automatically learned taxonomies it does not make sense to calculate the semantic cotopy of leaf nodes. On the contrary, we need to evaluate the internal nodes that group the leaf nodes. Let us assume the following notation:

T_A = *automatically learned taxonomy*
η_i = *node in a taxonomy*
$C(T_A)$ = *internal nodes + leaf nodes of* T_A
$I(T_A)$ = *internal nodes of* T_A
T_G = *gold standard taxonomy*
$C(T_G)$ = *internal nodes + leaf nodes of* T_G
$I(T_G)$ = *internal nodes of* T_G
$hyper(\eta_i)$ = *supernodes of* η_i *excluding the root*
$hypo(\eta_i)$ = *subnodes of* η_i *including* η_i
For $\eta_i \in I(T_A)$, the semantic cotopy is defined as:
$SC'(\eta_i) = (hyper(\eta_i) \cup hypo(\eta_i)) \cap C(T_G)$
For $\eta_i \in C(T_G)$, the semantic cotopy is defined as:
$SC''(\eta_i) = (hyper(\eta_i) \cup hypo(\eta_i))$

$$P(\eta_i, \eta_j) = \frac{|SC'(\eta_i) \cap SC''(\eta_j)|}{|SC'(\eta_i)|} \quad (1)$$

$$R(\eta_i, \eta_j) = \frac{|SC'(\eta_i) \cap SC''(\eta_j)|}{|SC''(\eta_j)|} \quad (2)$$

$$F(\eta_i, \eta_j) = \frac{2P(\eta_i, \eta_j)R(\eta_i, \eta_j)}{P(\eta_i, \eta_j) + R(\eta_i, \eta_j)} \quad (3)$$

Precision, recall and harmonic mean of node $\eta_i \in I(T_A)$ with respect to node $\eta_j \in C(T_G)$ are defined in Equations 1, 2 and 3. The F-score, FS, of node $\eta_i \in I(T_A)$ is the maximum F attained at any $\eta_j \in C(T_G)$ ($FS(\eta_i) = \text{argmax}_j F(\eta_i, \eta_j)$). Finally, the similarity TS of the entire taxonomy to the gold standard taxonomy is the average of the F-scores of each $\eta_i \in I(T_A)$ (Equation 4). The $TS(T_A, T_G)$ in Figure 1 is 0.9. All nodes of T_A have a perfect match, apart from *New Cluster 0* and *New Cluster 2*, which are matched against *computer network* and *meshwork* respectively, having a perfect precision but a lower recall since the cotopies of *computer network* and *meshwork* consist of three concepts. The automatically learned taxonomy has two redundant clusters that decrease its similarity.

$$TS(T_A, T_G) = \frac{1}{|I(T_A)|} \sum_{\eta_i \in I(T_A)} FS(\eta_i) \quad (4)$$

The similarity measure $TS(T_A, T_G)$ provides the similarity of the automatically learned taxonomy to the gold standard one, but it is not symmetric. Calculating the taxonomic similarity one way might not provide accurate results, in cases where T_A misses senses of the gold standard. This is due to the fact that we would only evaluate the internal nodes of T_A, partially ignoring the fact that T_A might have missed some parts of the gold standard taxonomy. For that reason, we also calculate $TS(T_G, T_A)$ which provides the similarity of the gold standard taxonomy to the automatically learned one. Finally, taxonomic similarities are combined to produce their harmonic mean (Equation 5).

$$TxSm(T_A, T_G) = \frac{2TS(T_G, T_A)TS(T_A, T_G)}{TS(T_G, T_A) + TS(T_A, T_G)} \quad (5)$$

4.3 Evaluation datasets & setting

The first gold standard taxonomy is derived by extracting from WordNet all the hyponyms of the senses of the word *network*. The extracted taxonomy contains 29 senses linguistically realized by 24 word sets (one sense might be expressed with more than one words), since *network* has 5 senses and *reseau* has 2 senses in the gold standard taxonomy. Note that we have disregarded senses only expressed by multiword expressions. The average polysemy of words is around 1.7. The second taxonomy is derived by extracting the concepts under the senses of the word *speaker*. The *speaker* taxonomy contains 52 senses linguistically realized by 50 word sets, since *speaker* has 3 senses included in the taxonomy. The average polysemy of words is around 1.58.

To create our datasets[3] we use the *Yahoo!* search api[4]. For each word w in each of the datasets, we is-

[3] Available in http://www.cs.york.ac.uk/aig/projects/indect/taxlearn

[4] http://developer.yahoo.com/search/ [Accessed:10/06/2009]

Parameter	Range
G^2 threshold (p_1)	5,10
Collocation frequency (p_2)	4,6,8
Collocation weight (p_3)	0.1,0.2,0.3,0.4

Table 3: Chosen parameters for the KM WSI method.

sue a query to *Yahoo!* that contains w and we download a maximum of 1000 pages. In cases where a particular sense is expressed by more than one word, the query was formulated by including all the words and putting the keyword *OR* between them. For each page we extracted fragments of text (paragraphs) that occur in <p> </p> html tags. We extracted 58956 and 78691 paragraphs for the *network* and *speaker* dataset respectively. The reason we extracted on average less content for the second dataset was that *Yahoo!* provided a small number of results for rare words such as *alliterator*, *anecdotist*, etc.

Table 3 shows the parameter ranges for the WSI method. Our method is evaluated according to these parameters. Our first baseline is *RAND*, which performs a random hierarchical clustering of senses to produce a binary tree. In each iteration two clusters are randomly chosen and form a new cluster, until we end up with one cluster taken to be the root. The performance of *RAND* is calculated by executing the random algorithm 10 times and then averaging the results. The second baseline is the taxonomy most frequent sense baseline (*TL MFS*), in which we do not perform WSI. Instead, given a parameter setting and a word w, all the collocations of w are grouped into one vector, which will possibly be dominated by collocations related to the MFS of w. WordNet mapping takes place and finally HAC with average-linkage is applied to create the taxonomy.

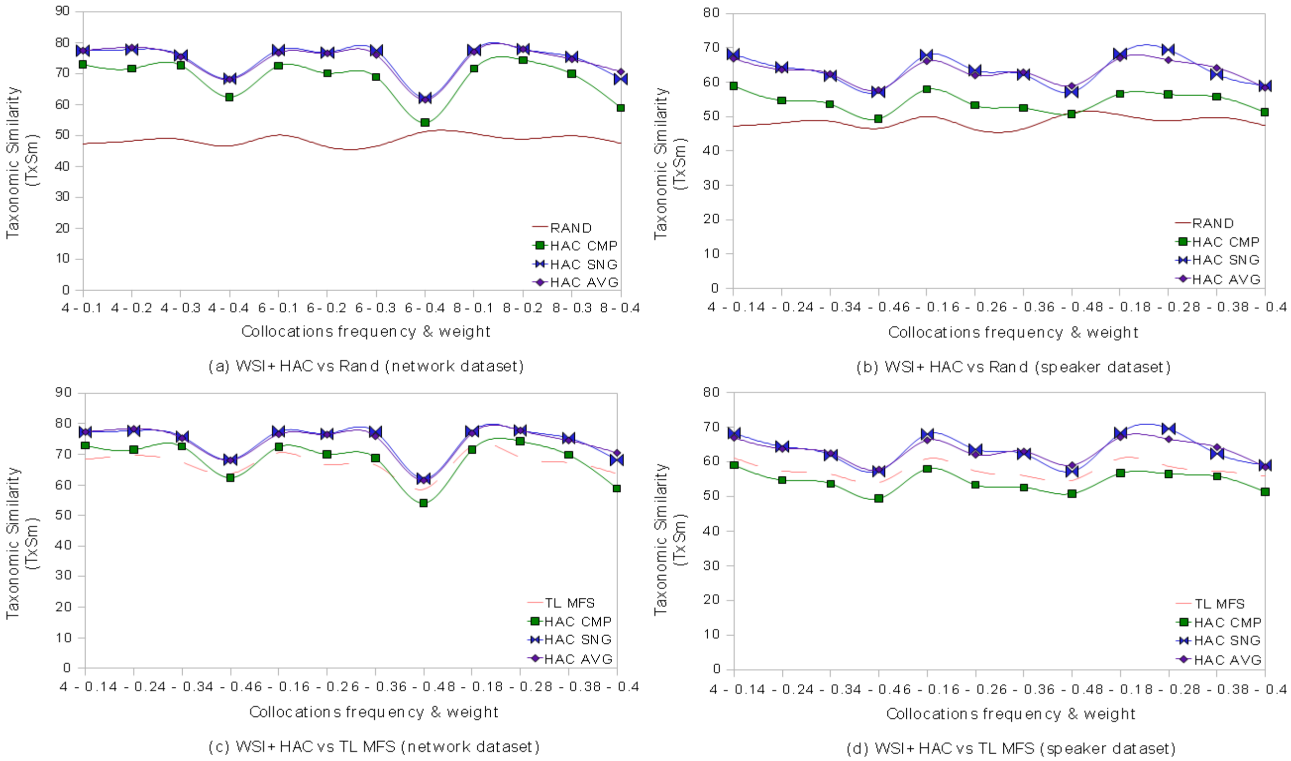
4.4 Results & discussion

Figures 4 (a) and 4 (b) show the performance of *HAC* with single-linkage (*HAC SNG*), average-linkage (*HAC AVG*) and complete-linkage (*HAC CMP*) against *RAND* for $p_1 = 5$ and different combinations of p_2 and p_3. It is clear that *HAC SNG* and *HAC AVG* outperform *RAND* by very large margins under all parameter combinations. In the *network* dataset, both of them achieve their highest distance from *RAND* (27.84%) at $p_2 = 8$ and $p_3 = 0.2$. In the *speaker* dataset, their highest distance from *RAND* (20.97% and 19.63% respectively) is achieved at $p_2 = 4$ and $p_3 = 0.1$. *HAC CMP* performs worse than the other HAC versions, yet it clearly outperforms *RAND* in all but one parameter combinations ($p_1 = 5$, $p_2 = 6$, $p_3 = 0.4$) in the *speaker dataset*.

Generally, for collocation weight equal to 0.4 the performance of all HAC versions drops. At this high collocation weight the WSI method produces a larger number of small clusters than in lower thresholds. This issue negatively affects both the mapping process and HAC. For example in the *speaker* dataset, for $p_1 = 5$, $p_2 = 8$ and $p_3 = 0.1$ our taxonomies contained 86.54% of the gold standard taxonomy senses. Increasing the collocation weight to 0.2 did not have any effect, but increasing the weight to 0.3 and then 0.4 led to 71.15% and 65.38% sense coverage. Overall, our conclusion is that all HAC versions exploit the WSI method and learn useful information better than chance. The picture is the same for $p_1 = 10$.

Figures 4 (c) and 4 (d) show the performance of HAC versions against the *TL MFS* baseline in the same parameter setting as above. We observe that both *HAC SNG* and *HAC AVG* perform significantly better than *TL MFS* apart from $p_3 = 0.4$, in which case all *HAC* versions perform worse. In the *network* dataset, the largest performance difference for *HAC SNG* is 10.12% and for *HAC AVG* 9.9% at $p_2 = 6$ and $p_3 = 0.2$. In the *speaker* dataset, the largest performance difference for *HAC SNG* is 10.83% and for *HAC AVG* 7.83% at $p_2 = 8$ and $p_3 = 0.2$. *HAC CMP* performs worse than *TL MFS* under most parameter settings in both datasets. The picture is the same for $p_1 = 10$.

Overall, the analysis of the WSI-based taxonomy learning approach against *TL MFS* shows that *HAC SNG* and *HAC AVG* perform better than *TL MFS* under all parameter combinations for both datasets. The main reason for their superior performance is that their learned taxonomies contain a higher number of senses than *TL MFS* as a result of the sense induction process. This greater sense coverage leads to the discovery of a higher number of correct taxonomic relations between senses than *TL MFS*, hence in a better performance. To conclude, our results verify our hypothesis and suggest that the unsupervised learning of word senses contributes to producing taxonomies with a higher similarity to the gold standard ones than traditional distributional similarity methods.



(a) WSI+HAC vs Rand (network dataset)

(b) WSI+HAC vs Rand (speaker dataset)

(c) WSI+HAC vs TL MFS (network dataset)

(d) WSI+HAC vs TL MFS (speaker dataset)

Figure 4: Performance analysis of the proposed method for $p_1 = 5$ and different combinations of p_2 and p_3.

Despite that, our evaluation also shows that in most cases *HAC CMP* is unable to exploit the induced senses and performs worse than *TL MFS*, *HAC SNG* and *HAC AVG*. This result was not expected, since *HAC SNG* employs a local criterion to merge two clusters and does not consider the global structure of the clusters, in effect, being biased towards elongated clusters. The observation of the gold standard taxonomies shows that they consist both of cohyponym concepts which are expected to be contextually related, but also of cohyponyms which are not expected to appear in similar contexts. For example, someone would expect a high similarity between *WAN*, *LAN*, or between *snood* and *tulle*. However, the same does not apply for *snood* and *cheesecloth* or *tulle* and *grillwork*, because *cheesecloth* and *grillwork* appear in significantly different contexts than *snood* and *tulle*. Despite that, all of them are cohyponyms. This issue is more prevalent in the *speaker* dataset, where concepts such as *loudspeaker*, *tannoy*, *woofer* are expected to be contextually related, while cohyponyms such as *whisperer*, *lecturer* and *interviewer* are not. This means that the gold standard taxonomies include elongated clusters and explains the superior performance of *HAC SNG*. This issue is not affecting *HAC AVG*, but it has a significant effect on *HAC CMP*. Generally, *HAC CMP* employs a non-local criterion by considering the diameter of a candidate cluster. This results in compact clusters with small diameters, as opposed to elongated ones.

5 Conclusion

We presented an unsupervised method for taxonomy learning that employs WSI to identify the senses of target words and then builds a taxonomy of these senses using HAC. We have shown that dealing with polysemy by means of sense induction helps to develop taxonomies that capture a higher number of correct taxonomic relations than traditional distributional similarity methods, which associate each target word with one vector of features, in effect, merging its senses.

Acknowledgements

This work is supported by the European Commission via the EU FP7 INDECT project, Grant No. 218086, Research area: SEC-2007-1.2-01 Intelligent Urban Environment Observation System.

References

E. Agirre and A. Soroa. 2007. SemEval-2007 Task 02: Evaluating Word Sense Induction and Discrimination Systems. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 7–12, Prague, Czech Republic.

R. A. Amsler. 1981. A Taxonomy for English Nouns and Verbs. In *Proceedings of the 19th ACL Conference*, pages 133–138, Stanford, California.

C. Biemann. 2006. Chinese Whispers - An Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems. In *Proceedings of TextGraphs*, pages 73–80, New York,USA.

P. Buitelaar, D. Olejnik, and M. Sintek. 2004. A Ptotégé Plug-in for Ontology Extraction from Text Based on Linguistic Analysis. In *Proceedings of the 1st European Semantic Web Symposium*, pages 31–44, Crete, Greece. CEUR-WS.org.

S. A. Caraballo. 1999. Automatic Construction of a Hypernym-labeled Noun Hierarchy from Text. In *Proceedings of the 37th ACL Conference*, pages 120–126, College Park, Maryland.

P. Cimiano, A. Hotho, and S. Staab. 2004. Comparing Conceptual, Divisive and Agglomerative Clustering for Learning Taxonomies from Text. In *Proceedings of the 16th ECAI Conference*, pages 435–439, Valencia, Spain.

P. Cimiano, A. Hotho, and S. Staab. 2005. Learning Concept Hieararchies from Text Corpora Using Formal Concept Analysis. *Journal of Artificial Intelligence Research*, 24:305–339.

P. Cimiano. 2006. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

T. Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1):61–74.

D. Faure and C. Nédellec. 1998. A Corpus-based Conceptual Clustering Method for Verb Frames and Ontology Acquisition. In *LREC workshop on Adapting lexical and corpus resources to sublanguages and applications*, pages 5–12, Granada, Spain.

C. Fellbaum. 1998. *Wordnet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts, USA.

B. Ganter and R. Wille. 1999. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. Translator-C. Franzke.

Z. Harris. 1968. *Mathematical Structures of Language*. Wiley, New York, USA.

M. A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th Coling Conference*, pages 539–545, Nantes, France.

C. H. Hwang. 1999. Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information. In *Proceedings of the 6th International Workshop on Knowledge Representation Meets Databases*, pages 14–20, Linkoping, Sweden. CEUR-WS.org.

B. King. 1967. Step-wise Clustering Procedures. *Journal of the American Statistical Association*, 69:86–101.

I. P. Klapaftis and S. Manandhar. 2008. Word Sense Induction Using Graphs of Collocations. In *Proceedings of the 18th ECAI Conference*, pages 298–302, Patras, Greece. IOS Press.

A. Maedche and S. Staab. 2002. Measuring Similarity between Ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW)*, pages 251–263, London,UK. Springer-Verlag.

D. Moldovan and A. Novischi. 2002. Lexical Chains for Question Answering. In *Proceedings of the 19th Coling Conference*, pages 1–7, Taipei, Taiwan.

R. Navigli and P. Velardi. 2004a. Learning Domain Ontologies from Document Warehouses and Dedicated web Sites. *Computational Linguistics*, 30(2):151–179.

R. Navigli and P. Velardi. 2004b. Structural Semantic Interconnection: a Knowledge-based Approach to Word Sense Disambiguation. In *Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 179–182, Barcelona, Spain.

M.L. Reinberger and P. Spyns. 2004. Discovering Knowledge in Texts for the Learning of Dogma-inspired Ontologies. In *Proceedings of the ECAI Workshop on Ontology Learning and Population*, pages 19–24, Valencia, Spain.

M. L. Reinberger, P. Spyns, W. Daelemans, and R. Meersman. 2003. Mining for Lexons: Applying Unsupervised Learning Methods to create ontology bases. In *CoopIS/DOA/ODBASE*, pages 803–819.

D. Sánchez and A. Moreno. 2005. Web-scale Taxonomy Learning. In *Proceedings of the Workshop on Learning and Extending Ontologies by using Machine Learning methods*, pages 53–60, Bonn, Germany.

P. H. A. Sneath and R. R. Sokal. 1973. *Numerical Taxonomy, The Principles and Practice of Numerical Classification*. W. H. Freeman, San Francisco, USA.