Integrating Domain and Paradigmatic Similarity for Unsupervised Sense Tagging

Roberto Basili\textsuperscript{1} and Marco Cammisa\textsuperscript{1} and Alfio Massimiliano Gliozzo\textsuperscript{2}

Abstract. An unsupervised methodology for Word Sense Disambiguation, called Dynamic Domain Sense Tagging, is presented. It relies on the convergence of two very well known unsupervised approaches (i.e. Domain Driven Disambiguation and Conceptual Density). For each target word a domain is dynamically modeled by expanding the its topical context, i.e. a set of words evoking the underlying/implicit domain where the word is located. The estimation of the paradigmatic similarity within such a specific lexicon is assumed as a disambiguation model. The Conceptual Density measure is here used to account for paradigmatic associations, and the top scored senses of the target word are selected accordingly. Results confirm the impact of domain based representation in capturing useful paradigmatic generalizations, especially when small text fragments are available. In addition, the precision/recall tradeoff of the resulting method can be tuned in a meaningful way, allowing us to achieve impressively high precision scores in a purely unsupervised setting.

1 Introduction

Semantic disambiguation is a relevant problem in a number of Artificial Intelligence applications, ranging from Ontology Engineering to Information Retrieval. Any text-driven ontology population or ontology mapping task, for instance, is implicitly a sense disambiguation problem at various degrees of grain and complexity. Another important application is query translation and expansion for cross-language Information Retrieval.

Unfortunately, Word Sense Disambiguation (WSD) is a corner stone in Natural Language Processing. In fact, the specificity of the target application domains and the involvement of languages different from English do not allow to rely on existing sense tagged resources, without whom state-of-the-art supervised approaches to WSD [6] cannot be adopted. Unsupervised approaches for semantic disambiguation thus play a central role in most of the above scenarios.

Among those approaches, [9] proposes a method to estimate predominant senses via a large untagged corpus, reaching high accuracy on standard WSD benchmarks. The underlying assumption is that predominant senses are not stable across domains and corpora, and that the notion of predominance can be modeled through an untagged corpus based analysis. The main limitation of that approach is that the sense distribution is modeled once for all on the basis of the corpus adopted for unsupervised learning, while word senses are first of all disambiguated by local constraints. The concept of predominance provide only an "a-priory" information about the probability distribution of word senses, that should be anyway refined by looking at a local context.

A possibility to merge both approaches is then to adopt the global information provided by the whole corpus to the local constraints provided by tokens in the surrounding context of the word to be disambiguated. Each sentence in fact suggests a topical context where the semantics of the target word is highly constrained. The modeling of such local domain evidence can thus exploit several sources of semantic evidences. Words describing a topical contexts of the target word \textit{tw} can be used to impose tighter constraints to the interpretation of \textit{tw}. They express indeed a variety of information:

- \textbf{Domain information}, as the semantic domain of the context can be used to discriminate among metaphorical sense shifting (e.g. \textit{virus} in Biology vs. Computer science)
- \textbf{Paradigmatic information}, as important siblings or synonyms \textit{w} of \textit{tw} (e.g. \textit{bugs} vs. \textit{bacteria}) are likely to appear in topical-contexts: class information for sense distinctions is thus made available

In this paper we investigate a disambiguation model that captures the above idea within an unsupervised framework aiming to address the semantic disambiguation problem of classical AI tasks, e.g. query translation and ontology learning. The model, named Dynamic Domain Semantic Tagging (DDST), relies on unsupervised learning techniques and Wordnet [11]. The DDST algorithm integrates two well known methodologies, i.e. the Domain Driven Disambiguation (DDD) [8] and the similarity metric known as Conceptual Density (CD) [1], reflected in the following two steps:

\begin{itemize}
  \item \textbf{Domain Detection} Extract the topical-context for the target word.
  \item \textbf{Paradigmatic Disambiguation} Estimate the paradigmatic cohesion of the domain lexicon and then select the sense maximizing such an association within the topical context.
\end{itemize}

Details of each individual step will be provided in Section 2 and Section 3, respectively.

The DDST approach solves the main limitations of the CD algorithm: the lack of semantic relations among concepts having different ontological types (e.g. the nouns \textit{athlete} and \textit{sport} are totally unrelated in WordNet, because they belong to the disjointed hierarchies of \textit{life forms} and \textit{act}, respectively). To this aim, the DDST algorithm identifies a wider context (i.e. the semantic domain) for the word to be disambiguated. In this way a wider region of WordNet is involved, providing the CD algorithm of a more reliable evidence.

The DDST algorithm is then totally unsupervised, because it does not rely on the availability of sense tagged data and it does not exploit neither a prior estimation of sense frequency distribution [9] nor the
sense ordering provided by WordNet (reflecting the sense predominance in Senseval). It can be thus applied to a large set of tasks and languages once a WordNet like lexical database and a large corpus have been provided. It represents a technologically viable solution, uniform across different languages and application scenarios.

The result of the Domain Detection step is a sensible improvement in the accuracy of the overall lexical disambiguation process. To show this, in Section 4 we compared the DDST performances against a classical CD algorithm over the Senseval-3 English all words task [10]. First, the DDST algorithm achieves a substantial improvement against a random baseline. More importantly, its superiority is particularly evident when small contexts of the target word are taken into account, and this confirms its applicability to query disambiguation (expansion or translation). Conclusion and future work are finally discussed in Section 5

2 Dynamic Generation of Domain Lexicons

The Domain Detection step is inspired by the Domain Driven Disambiguation (DDD) methodology for WSD, originally proposed by [8]. DDD is a supervised approach to WSD consisting in selecting the word sense that maximizes the similarity with the domain of the context. In its original formulation the DDD methodology required (i) a knowledge base containing domain information for each sense of the target word to be disambiguated and (ii) an automatic algorithm to recognize the domain of the context in which the word occurs. To accomplish the first requirement authors proposed the exploitation of WordNet Domains [7], an extension of WordNet, where each synset has been manually annotated by domain labels selected from a predefined domain set. Unfortunately, this approach is very restrictive for the following reasons:

1. The manual development of lexical resources is very expensive.
2. The domain set is fixed once for all in the resource, while the relevant domain distinctions are usually task and context dependent.
3. Domain information alone is not sufficient to resolve lexical ambiguity: syntagmatic and paradigmatic aspects of sense distinction are not taken into account.

The Domain Detection step is an attempt to solve both the first and the second problems reported above, defining a dynamic approach to identify only those domain distinctions actually involved in the application domain without requiring any manual domain annotation. In fact, in the perspective proposed by the authors, domain distinctions have been conceived as a "static" notion, reflecting the lexicographers’ intuition of an a-priori organization of the world. On the other hand, we believe that such a notion is strongly dependent on the specificity of the particular discourse or inference we are dealing with, preventing us to define such a static domain model.

For these reasons we adopted the more flexible notion of topical-context to deal with the domain modeling problem. Topical context are collections of words, describing the domain in which the text is located. They can be determined in a totally unsupervised way by adopting term/text similarity techniques, and can be used to plug "external knowledge" from a large corpus into the disambiguation process.

Such a similarity metric can be provided by defining a Domain Space [5], where both terms and texts are represented by means of Domain Vectors (DVs). We exploit the duality property of the Domain Space [5], where both terms and texts are represented by means of Domain Vectors (DVs). Such a similarity metric can be provided by defining a Domain Space [5], where both terms and texts are represented by means of Domain Vectors (DVs). Such a similarity metric can be provided by defining a Domain Space [5], where both terms and texts are represented by means of Domain Vectors (DVs).

### Figure 1

Output of the domain discovery step for the sentence $c = \text{"He took a walk along the bank of the river"}$. The topical-context for this occurrence of bank, contains the terms river, hill, gate, snow and road, among the others (see Figure 1).

The Domain Space can be defined once a Domain Model (DM) is available. A DM is represented by a $k 	imes k'$ rectangular matrix $D$, containing the domain relevance for each term with respect to each domain, where $k$ is the cardinality of the vocabulary, and $k'$ is the size of the Domain Set. Once a DM has been defined by the matrix $D$, the Domain Space is a $k'$-dimensional space, in which both texts and terms are associated to Domain Vectors (DVs), i.e. vectors representing their domain relevances with respect to each domain. The DV $\vec{w}_i$ for the term $w_i \in V$ is the $i^{th}$ row of $D$, where $V$ is the vocabulary of the corpus. The similarity among DVs in the Domain Space is estimated by means of the cosine operation.

To show this, we illustrate the Domain Detection process by an example. Suppose that the word bank has to be disambiguated into the sentence "He took a walk along the bank of the river". The topical-context for this occurrence of bank, contains the terms river, hill, gate, snow and road, among the others (see Figure 1).

The Domain Space can be defined once a Domain Model (DM) is available. A DM is represented by a $k 	imes k'$ rectangular matrix $D$, containing the domain relevance for each term with respect to each domain, where $k$ is the cardinality of the vocabulary, and $k'$ is the size of the Domain Set. Once a DM has been defined by the matrix $D$, the Domain Space is a $k'$-dimensional space, in which both texts and terms are associated to Domain Vectors (DVs), i.e. vectors representing their domain relevances with respect to each domain. The DV $\vec{w}_i$ for the term $w_i \in V$ is the $i^{th}$ row of $D$, where $V$ is the vocabulary of the corpus. The similarity among DVs in the Domain Space is estimated by means of the cosine operation.

DVs can be acquired from texts in a totally unsupervised way by exploiting a lexical-coherence assumption [5]. To this aim, term clustering algorithms can be adopted: each cluster represents a Semantic Domain and the degree of association among terms and clusters, estimated by the learning algorithm, provides a domain relevance function. For our experiments we adopted a clustering strategy based on Latent Semantic Analysis (LSA), following the methodology described in [6]. This operation is done off-line, and can be efficiently performed on large corpora. To filter out noise, we included only those terms having a frequency higher than 5 in the corpus.

When a word $w_i$ is located into the context $c = (w_{-h}, \ldots, w_{-1}, w_0, w_{+1}, \ldots, w_{+h})$, has to be disambiguated, first the DV $\vec{c}$ of its context is estimated by

$$\bar{c} = \sum_{i=-h}^{+h} \vec{w}_i$$  

$\vec{c}$ defines a neighbourhood including the DVs $\vec{w}_i$ of other corpus terms $w_i \in V$, topically associated to the source text. A topical context for the $\vec{c}$ can be thus formally defined by: $\{w \in V \mid \| \vec{c} - \vec{w} \| < \theta \}$, where $\|$ is any valid norm in the LSA space derived from the cosine similarity and $\theta \in \mathbb{R}$. The topical-context includes all those terms whose distance (i.e. similarity) with the context $\vec{c}$ is below (above) a domain specificity threshold. In general, the lower the domain threshold $\theta$, the higher the specificity of the returned topical-context and the smaller its size. Another restriction that can be imposed in the modeling of a proper topical-context is polysemy. In fact, ambiguous words inherently activate noisy generalizations during the similarity estimations due to their spurious sense(s). On the other hand, monosemous words should always refer to the unique correct sense, thus activating the "right" paradigmatic associations.

---

3 The development of WordNet Domains took about 2 man year.
In order to control such phenomenon, those words whose polysemy is over a predefined threshold \( \rho \) can be filter out. A formal definition for a topical-context in the disambiguation perspective is thus given by:

\[
TC(c) = \{ w \in \mathcal{V} \mid \| \vec{v} - \vec{c} \| < \theta \text{ and } \text{pol}(w) < \rho \}
\]

where \( \text{pol}(w) \) expresses the polysemy (i.e. number of senses) of \( w \) in WordNet.

\[\text{Figure 2. Conceptual Density for the words bank, walk and river} \]

\[\text{Figure 3. Conceptual Density for the word bank together with its topical-context}\]

3 Semantic Disambiguation through Conceptual Density

In [2] an \( n \)-ary similarity measure aimed to support an unsupervised approach to semantic tagging has been presented. It represents a variant of the notion of *Conceptual Density* previously suggested as a tool for sense disambiguation [1]. In [3] it is applied for learning sense preferences in Wordnet from available syntactic patterns found in a corpus. Sets of syntactically similar words are clustered and the best senses \( \alpha \) as useful semantic explanations of the underlying syntactic phenomena are selected. In a Bayesian perspective, rules for disambiguation (i.e. estimates for \( \text{prob}(\alpha | r, w) \)) are derived from similar syntactic contexts \( r \), i.e. the sets of different words \( w \) in the same grammatical contexts. In order to favour some senses an estimation of semantic similarity "local" to the contexts \( r \) is carried out via the conceptual density \( (\text{cd}^*\alpha) \) measure.

The same definition of conceptual density as in [3] is proposed here for the semantic disambiguation phase. However, a different notion of context \( r \) is applied. Instead of using syntactic equivalence (i.e. occurrence in the same syntactic contexts), we define \( r \) as the neighbourhood \( TC(c) \) determined by Eq. 2 for the source context \( c \). The hypothesis is that some members in a topical context \( TC(c) \) are expected to share paradigmatic properties with the target word \( tw \), their sense in \( TC(c) \) is thus semantically close to the suitable sense of the \( tw \) in the source context \( c \). The closest sense \( \alpha \) is thus selected by maximizing the conceptual density of the Wordnet hierarchy rooted at \( \alpha \).

As illustrated by Figure 3, the topical-context provide more evidence than the mere list of the words actually located in context of the word to be disambiguated (i.e. walk and river, see Figure 2) improving the final disambiguation. In fact, the (wrong) generalization for the word bank provided by the classical CD algorithm is *accomplishment achievement*, while if the topical-context is taken into account the CD algorithm select the (correct) sense of incline slope land.

3.1 A similarity measure for large scale semantic disambiguation

In Section 2 a topical context \( TC(c) \), i.e. a domain specific lexicon, originating from a textual context \( c \) has been defined via Eq. 2. Words \( w \in T(c) \) can be generalized through senses \( \alpha \) of the Wordnet hierarchy. The likelihood of a sense \( \alpha_{tw} \) for the target word \( tw \) is proportional to the number of other words \( w \in TC(c) \) that have common generalizations with \( tw \) along the paths activated by their \( \alpha \) in the hierarchy. A measure of the suitability of senses \( \alpha \) for words \( w \in TC(c) \) is thus the information density of the subtrees rooted at \( \alpha \). The higher is the number of nodes in the subhierarchy that generalize some nouns \( w \in TC(c) \), the better is the related interpretation \( \alpha \). The conceptual density models the former notion and provides a measure for the latter.

\[\text{DEF (Conceptual Density). Given a context } c \text{ for a target word } tw, \text{ its topical context } TC(c), \text{ a synset } \alpha \text{ in Wordnet used to generalize } n \text{ different nouns } w \in TC(c), \text{ the conceptual density, } \text{cd}^{TC(c)}(\alpha), \text{ of } \alpha \text{ with respect to } TC(c) \text{ is defined as:}\]

\[
\text{cd}^{TC(c)}(\alpha) = \frac{\sum_{i=0}^{h} \mu}{\text{area}(\alpha)}
\]

where:

- \( h \) is the estimation of the depth of a tree able to generalize the \( n \) nouns. Its actual value is estimated by:

\[
h = \begin{cases} 
\lfloor \log_{\mu} n \rfloor & \text{if } \mu \neq 1 \\
n & \text{otherwise}
\end{cases}
\]

- \( \mu \) is the average number of sons per node in the full Wordnet subhierarchy dominated by \( \alpha \). Its estimation is available statically from Wordnet and can be evaluated *a priori* without uncertainty. Notice that when nodes belong to unbalanced branches of the hierarchy, the value for \( \mu \) can approach (and in fact is) 1, so that a specific treatment of them is needed in the definition.

- \( \text{area}(\alpha) \) is the number of nodes in the \( \alpha \) subhierarchy. This value is also estimated statically from Wordnet.

Equation 3 applies to any valid and common generalization of the nouns \( w \in TC(c) \) as topically related to \( tw \). The aim of this method however is to reduce as much as possible the number of such generalizations. In [3] two properties have been defined that determine the best generalization set \( G(c) \) among all the valid generalization set \( S \) applicable to an originating context \( c \), i.e.

\[S = \{ \alpha | \alpha \text{ is an hyperonim of at least one sense } \sigma_w, w \in TC(c) \}\]
a) (Useful generalization) $S$ is a useful set of generalizations if $S$ covers the entire set $TC(c)$ and $\forall \alpha \in S$, $\alpha$ is a hypernym of at least two words $w_1 \neq w_2 \in TC(c)$.

b) (Maximally dense) The factor $\sum_{\alpha \in S} cd_{TC(c)}(\alpha)$ must be maximal, among the different $S$ in the family of useful sets of generalizations, i.e.

$$G(c) = \arg\max_S \sum_{\alpha \in S} cd_{TC(c)}(\alpha)$$

where $S$ satisfies a).

The $G(c)$ set characterized by a) and b) is the best paradigmatic interpretation of the topical context $TC(c)$ generated by $c$. In [3] a greedy algorithm able to efficiently compute the minimal set $G(c)$ that covers $TC(c)$ with a maximal density is defined\(^4\). The outcome is the minimal set $G(c)$ of synsets that are the maximally dense generalizations of at least two words in $TC(c)$. Words $w \in TC(c)$ for which no such generalization may exist and will not be represented in the resulting set $G(c)$. Scores $cd_{TC(c)}(\alpha)$ model the strength by which words $w \in TC(c)$ are related to senses $\alpha$, as suitable paradigmatic explanations of the topical context $TC(c)$.

The semantic disambiguation of a target word $tw$ in a context $c$ depends on the subset of generalizations $\alpha \in G(c)$ concerning some of its senses $\sigma_{tw}$. Let $G_{tw}(c)$ be such a subset, i.e.

$$G_{tw}(c) = \{ \alpha \in G(c) \mid \exists \sigma_{tw} \text{ such that } \sigma_{tw} < \alpha \}$$

(5)

where $<$ denotes the transitive closure of the hyponymy relation in Wordnet.

DEF (Semantic disambiguation). Given a context $c$ for a target word $tw$ and the generalization set $G_{tw}(c)$ defined above, the set $\sigma(tw,c)$ of correct sense(s) for $tw$ in $c$ is given by:

$$\sigma(tw,c) = \{ \sigma_{tw} \mid \sigma_{tw} < \pi \}$$

(6)

with $\pi = \arg\max_{\alpha \in G_{tw}(c)} cd_{TC(c)}(\alpha)$. Notice that when $G_{tw}(c)$ is empty then $\sigma(tw,c) = \emptyset$ so that no decision can then be taken for the underlying context $c$. Other disambiguation heuristics should be applied. Moreover, it is also possible that $|\sigma(tw,c)| > 1$ so that multiple senses may be assigned to a context $c$. Equation 6 thus defines an (unsupervised) semantic disambiguation model. It will be evaluated in the next section.

4 Evaluation

In order to show the benefits obtained from adding the topical-context words, we contrasted the DDST with a classical CD algorithm over a WSD benchmark. This evaluation helps in the objective assessment of the method, but it focuses on just one of the variety of tasks to which DDST applies.

In this section we first describe the details of the implementation of the DDST algorithm and the adopted WSD benchmark (Subsection 4.1). Then, we contrast the performances of the DDST algorithm with those of a classical CD approach to WSD (Subsection 4.2). Finally, Subsection 4.3 investigates the precision/recall trade-off.

4.1 Experimental Settings

For the experiments, we used a C++ implementation of the CD algorithm described in Subsection 3, developed at the University of Rome Tor Vergata. Independently from the target task, the input of the tool is a set of nouns and the output is a set of generalization synsets from WordNet 1.7.1 with the related CD values. For the WSD purposes, we concentrated just on the CD estimation of the target words. The tool also returns a probability estimation (based on the CD score) for each sense of the input words. In the experiments The sense of the target word with maximal probability is selected as a final output. The DM has been acquired from the British National Corpus [4], following the methodology described in [6].

Results reported in the following subsections have been evaluated on the Senseval-3 English All-Words task [10]. This task consists on tagging all the words contained in tree middle sized English texts (i.e. about 1500 words each). As the CD algorithm depends on the topology of the lexical hierarchy, DDST applies best to nouns. Then, for our experiments, we concentrated on the subset of the target words made by the 876 nouns. The Senseval tasks also provides an automatic scoring procedure for precision and recall. As a baseline for the task, we implemented a random WSD algorithm, by assigning a randomly chosen sense to each target word, achieving F1=0.39\(^6\). One peculiarity of the Senseval-3 task is that wide contexts are made available to the WSD algorithm. However, in many practical applications (e.g. the disambiguation of IR queries) this assumption is unrealistic, therefore in this paper we will try to minimize the context window to be taken into account for disambiguation as much as possible.

A preliminary investigation about the role of the underlying parameters has been carried. No significant variations are observed\(^6\). The following setting is thus adopted for the dimensionality, similarity and polisemyc control, respectively: $k'=100$, $\theta=0.5$, $\rho=2$.

4.2 Role of the Topical Context

To demonstrate the usefulness of the Domain Detection step, and the effectiveness of the proposed methodology, we compared the DDST results to those of a classical CD algorithm on the Senseval-3 task. Results are reported in Figure 4, where the two algorithms have been compared by varying the context size (i.e. $h = \{5, 10, 20, 50\}$). DDST is largely above the random baseline for every test. DDST and CD converges asymptotically when the context size increase, while the former is always superior in our experiments. Improvements are impressive especially when small contexts are considered, increasing the accuracy from 29.45 to 52.05 when a window of $\pm 5$ tokens is considered. Surprisingly, the performances of DDST slightly decrease when wider contexts are considered, in contrast to CD. However, data show that topical contexts generated from smaller text fragments have performance sensibly higher than larger text portions (52% vs. 48%).

In general, all the experiments confirm that the implicit domain of a text is a fundamental aspect to capture lexical semantics. This aspect is better addressed by adopting unsupervised learning techniques to expand small contexts with topical-contexts rather than considering larger fragments of the originating document. The role of the topical contexts is then crucial for all those application scenarios where just small texts are available, as in most of the typical IA settings (e.g. dialogue systems, natural language interfaces).

\(^4\) A demo version of the algorithm and the conceptual density measure can be accessed at http://ai-nlp.info.uniroma2.it/Estimator/cd.htm

\(^5\) F1 is the harmonic mean between precision and recall, i.e. $F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

\(^6\) The difference between the F1 achieved by the best configuration an those achieved by the the worst is about 0.03.
4.3 Maximizing Precision

A relevant aspect to be taken into account when performing WSD is precision. In many cases, it would be preferable to tag only those words on which the algorithm is more confident, and reduce the tagging noise. For example, when WSD is used for query expansion, the wrong sense assignments would turn out in wrong expanding words, that result in misleading queries.

The DDST algorithm allows to define an effective selection criterion as its output can be considered as a probability distribution over the senses of the target word. This is done by normalizing the cumulative CD values of individual senses\(^7\) by the whole contribution. As this normalization gives rise to a probability estimate, then the entropy \((E(tw) = - \sum_{i=1}^{n} p_i(tw) \cdot \log(p_i(tw)))\) with \(p_i\) as the probability of the \(i\)-th sense being correct for \(tw\), and the perplexity, i.e. its exponential (i.e. \(pp(tw) = 2^{E(tw)}\)) can be measured. In particular the perplexity can be considered as an estimation of the confidence of the prediction provided by the DDST algorithm. The adopted selection criterion can thus be obtained by thresholding the perplexity of the output probability by a parameter \(\tau\).

Figure 5 reports Precision, Recall, F1 and Coverage of the system for different values of \(\tau\). showing that this parameter can be tuned meaningfully to control the trade-off between precision and recall. As shown in Fig. 5, \(\tau\) allows to capture accurate predictions. Precision is above 80% along with reasonable recall values (\(\tau \sim 0.1\)), while a coverage of about the 60% of the cases can be still achieved with a 70% of precision (\(\tau \sim 1.8\)).

5 Conclusion and Future Work

In this paper we proposed a fully unsupervised methodology for semantic disambiguation useful in a large set of application scenarios, such as cross language information retrieval, question answering and ontology learning. The overall model combines in a natural and effective way distributional information observable in a corpus (mainly domain-specific topical associations among words) with the topological (i.e. paradigmatic) properties of a semantic network. A relevant side-effect of the disambiguation step is capturing paradigmatic properties, projecting out of the hierarchy a meaningful subportion as a suitable interpretation local to the source context. Such dynamic “view” can be seen as a domain-specific ontology that better characterizes the target text fragment. Results clearly shows that our approach works very well when small contexts of the target words are taken into account (e.g. queries and glosses), and can be tuned in order to boost precision to very high scores (over 80%). This is a very important feature in many IA settings, such us ontology population. Although experiments have been run over a classical WSD task, they demonstrate the technological perspectives opened by the proposed method over typical complex tasks in modern AI. This paper is just a first attempt to provide a dynamic notion to the concept of semantic domain and its relation to the paradigmatic structure in WordNet. For the future, we plan to further extend this methodology to address several lacks of this research. In particular we plan to improve the CD algorithm by providing a corpus score to each synset and to integrate syntactic constraints in the disambiguation process. In addition we plan to exploit the proposed technique to address the traditional AI problem tuning and large scale ontology to a specific domain.

---

\(G_{tw}(c)\) may cover the same sense \(\sigma(tw, c)\) providing different evidence in favour of \(\sigma(tw, c)\).

\(^{7}\) Different generalizations in \(G_{tw}(c)\) may cover the same sense \(\sigma(tw, c)\) providing different evidence in favour of \(\sigma(tw, c)\).

REFERENCES


---

Figure 4. Accuracy varying the context size

Figure 5. Precision, Recall, F1-measure, Coverage for 100x100 LSA matrix with ±5 context at various thresholds \(\tau\)