Instance Based Lexical Entailment for Ontology Population

Claudio Giuliano and **Alfio Gliozzo**
FBK-irst, Istituto per la Ricerca Scientifica e Tecnologica
I-38050, Trento, ITALY
{giuliano,gliozzo}@itc.it

Abstract

In this paper we propose an instance based method for lexical entailment and apply it to automatic ontology population from text. The approach is fully unsupervised and based on kernel methods. We demonstrate the effectiveness of our technique largely surpassing both the random and most frequent baselines and outperforming current state-of-the-art unsupervised approaches on a benchmark ontology available in the literature.

1 Introduction

Textual entailment is formally defined as a relationship between a coherent text T and a language expression, the hypothesis H. T is said to entail H, denoted by $T \rightarrow H$, if the meaning of H can be inferred from the meaning of T (Dagan et al., 2005; Dagan and Glickman., 2004). Even though this notion has been recently proposed in the computational linguistics literature, it has already attracted a great attention due to the very high generality of its settings and to the indubitable usefulness of its (potential) applications.

In this paper, we concentrate on the problem of lexical entailment, a textual entailment subtask in which the system is asked to decide whether the substitution of a particular word w with the word e in a coherent text $H_w = H^l w H^r$ generates a sentence $H_e = H^l e H^r$ such that $H_w \rightarrow H_e$, where H^l and H^r denote the left and the right context of w, respectively. For example, given the word 'weapon' a system may substitute it with the synonym 'arm', in order to identify relevant texts that denote the sought concept using the latter term. A particular case of lexical entailment is recognizing synonymy, where both $H_w \rightarrow H_e$ and $H_e \rightarrow H_w$ hold.

In the literature, slight variations of this problem are also referred to as *sense matching* (Dagan et al., 2006), *lexical reference* (Glickman et al., 2006a) and *lexical substitution* (Glickman et al., 2006b). They have been applied to a wide variety of tasks, such as semantic matching, subtitle generation and Word Sense Disambiguation (WSD). Modeling lexical entailment is also a prerequisite to approach the SemEval-2007 lexical substitution task[1], consisting of finding alternative words that can occur in given context.

In this paper, we propose to apply an approach for lexical entailment to the ontology population task. The basic idea is that if a word entails another one in a given context then the former is an instance or a subclass of the latter. This approach is intuitively appealing because lexical entailment is intrinsically an unsupervised task, therefore it does not require lexical resources, seed examples or manually annotated data sets. Unsupervised approaches are particularly suited for ontology population, whose goal is to find instances of concepts from corpora, because both corpus and the ontology sizes can scale up to millions of documents and thousands of concepts, preventing us from applying supervised learning. In addition, the top level part of the ontology (i.e., the *Tbox* in the Description Logics terminology) is very

[1] http://nlp.cs.swarthmore.edu/semeval/tasks/task10/description.shtml

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 248–256, Prague, June 2007.

often modified during the ontology engineering lifecycle, for example by introducing new concepts and restructuring the subclass_of hierarchy according to the renewed application needs required by the evolution of the application domain. It is evident that to preserve the consistency between the Tbox and the *Abox* (i.e., the set of instances and their relations) in such a dynamic ontology engineering process, supervised approaches are clearly inadequate, as small changes in the TBox will be reflected into dramatic annotation effort to keep instances in the Abox aligned.

The problem of populating a predefined ontology of concepts with novel instances implies a WSD task, as the entities in texts are ambiguous with respect to the domain ontology. For example, the entity *Washington* is both the name of a state and the name of a city. In the ontology population settings traditional WSD approaches cannot be directly applied since entities are not reported into dictionaries, making the lexical entailment alternative more viable. In particular, we model the problem of ontology population as the problem of recognizing for each mention of an entity of a particular coarse-grained type (e.g., location) the fine-grained concept (e.g., lake or mountain) that can be substituted in texts preserving the meaning. For example, in the sentence "the first man to climb the Everest without oxygen", "Everest" can be substituted with the word *mountain* preserving the meaning, while the sentence is meaningless when "Everest" is replaced with the word *lake*. Following the lexical entailment approach, the ontology population task is transformed into the problem of recognizing the term from a fine-grained set of categories (e.g., city, country, river, lake and mountain) that can be substituted in the contexts where the entity is mentioned (e.g., Everest in the example above).

The main contributions of this paper are summarized as follows. First, we propose a novel approach to lexical entailment, called Instance Based Lexical Entailment (IBLE), that allows approaching the problem as a classification task, in which a given target word (i.e., the entailing word) in a particular context is judged to entail a different word taken from a (pre-defined) set of (possible) candidate entailed words (see Section 3). Second, we exploit the IBLE approach to model the ontology population task as follows. Given a set of candidate concepts belonging to generic ontological types (e.g., people or locations), and a set of pre-recognized mentions of entities of these types in the corpus (e.g., Newton, Ontario), we assign the entity to the class whose lexicalization is more frequently entailed in the corpus. In particular, as training set to learn the fine-grained category models, we use all the occurrences of their corresponding expressions in the same corpus (e.g., we collected all occurrences in context of the word *scientist* to describe the concept `scientist`). Then, we apply the trained model to classify the pre-recognized coarse-grained entities into the fine-grained categories.

Our approach is fully unsupervised as for training it only requires occurrences of the candidate entailed words taken in their contexts. Restricted to the ontology population task, for each coarse-grained entity (e.g., location), the candidate entailed words are the terms corresponding to the fine-grained classes (e.g., lake or mountain) and the entailing words are mentions of entities (e.g., New York, Ontario) belonging to the coarse-grained class, recognized by an entity tagger.

Experiments show that our method for recognizing lexical entailment is effective for the ontology population task, reporting improvements over a state-of-the-art unsupervised technique based on contextual similarity measures (Cimiano and Völker, 2005). In addition, we also compared it to a supervised approach (Tanev and Magnini, 2006), that we regarded as an upper bound, obtaining comparable results.

2 The Ontology Population Task

Populating concepts of a predefined ontology with instances found in a corpus is a primary goal of knowledge management systems. As concepts in the ontology are generally structured into hierarchies belonging to a common ontological type (e.g., people or locations), the problem of populating ontologies can be solved hierarchically, firstly identifying instances in texts as belonging to the topmost concepts, and then assigning them to a fine-grained class. Supervised named entity recognition (NER) systems can be used for accomplishing the first step. State-of-the-art NER systems are characterized by high accuracy, but they require a large amount of training data. However, domain specific ontologies generally contains many "fine-grained" categories (e.g., particular categories of people, such as writers, scientists, and so on) and, as a consequence, supervised methods cannot be used because the annotation costs would become prohibitive.

Therefore, in the literature, the fine-grained classification task has been approached by adopting weakly supervised (Tanev and Magnini, 2006; Fleischman and Hovy, 2002) or unsupervised methods (Cimiano and Völker, 2005). Tanev and Magnini (2006) proposed a weakly supervised method that requires as training data a list of terms without context for each class under consideration. Such list can be automatically acquired from existing ontologies or other sources (i.e., database fields, web sites like Wikipedia, etc.) since the approach imposes virtually no restrictions on them. Given a generic syntactically parsed corpus containing at least each training entity twice, the algorithm learns, for each class, a feature vector describing the contexts where those entities occur. Then it compares the new (unknown) entity with the so obtained feature vectors, assigning it to the most similar class. Fleischman and Hovy (2002) approached the ontology population problem as a classification task, providing examples of instances in their context as training examples for their respective fine-grained categories.

The aforementioned approaches are clearly inadequate to recognize such fine-grained distinctions, as they would require a time consuming and costly annotation process for each particular class, that is clearly infeasible when the number of concepts in the ontology scales up. Therefore, most of the present research in ontology population is focusing on either unsupervised approaches (Cimiano and Völker, 2005) or weakly supervised approaches (Tanev and Magnini, 2006).

Unsupervised approaches are mostly based on term similarity metrics. Cimiano and Völker (2005) assign a particular entity to the fine-grained class such that the contextual similarity is maximal among the set of fine-grained subclasses of a coarse-grained category. Contextual similarity has been measured by adopting lexico-syntactic features provided by a dependency parser, as proposed in (Lin, 1998).

3 Instance Based Lexical Entailment

Dagan et al. (2006) adapted the classical supervised WSD setting to approach the sense matching problem (i.e., the binary lexical entailment problem of deciding whether a word, such as *position*, entails a different word, such as *job*, in a given context) by defining a one-class learning algorithm based on support vector machines (SVM). They train a one-class model for each entailed word (e.g., all the occurrences of the word *job* in the corpus) and, then, apply it to classify all the occurrences of the entailing words (e.g., the word *position*), providing a binary decision criterion[2]. Similarly to the WSD case, examples are represented by feature vectors describing their contexts, and then compared to the feature vectors describing the context of the target word.

In this paper, we adopt a similar strategy to approach a multi-class lexical entailment problem. The basic hypothesis is that if a word w entails e in a particular context ($H_w \rightarrow H_e$), then some of the contexts T_e^j in which e occurs in the training corpus are similar to H_w. Given a word w and an (exhaustive) set of candidate entailed words $E = \{e_1, e_2, \ldots, e_n\}$, to which we refer hereafter with the expression "substitution lexica", our goal is to select the word $e_i \in E$ that can be substituted to w in the context H_w generating a sentence H_e such that $H_w \rightarrow H_e$. In the multi-class setting, supervised learning approaches can be used. In particular, we can apply a one-versus-all learning methodology, in which each class e_i is trained from both positive (i.e., all the occurrences of e_i in the corpus) and negative examples (i.e., all the occurrences of the words in the set $\{e_j | j \neq i\}$).

Our approach is clearly a simplification of the more general lexical entailment settings, where given two generic words w and e, and a context $H = H^l w H^r$, the system is asked to decide whether w entails e or not. In fact, the latter is a binary classification problem, while the former is easier as the system is required to select "the best" option among the substitution lexicon. Of course providing such set could be problematic in many cases (e.g., it could be incomplete or simply not available for

[2]This approach resembles the pseudo-words technique proposed to evaluate WSD algorithms at the earlier stages of the WSD studies (Gale et al., 1992), when large scale sense tagged corpora were not available for training supervised algorithms.

many languages or rare words). On the other hand, such a simplification is practically effective. First of all, it allows us to provide both positive and negative examples, avoiding the use of one-class classification algorithms that in practice perform poorly (Dagan et al., 2006). Second, the large availability of manually constructed substitution lexica, such as WordNet (Fellbaum, 1998), or the use of repositories based on statistical word similarities, such as the database constructed by Lin (1998), allows us to find an adequate substitution lexicon for each target word in most of the cases.

For example, as shown in Table 1, the word *job* has different senses depending on its context, some of them entailing its direct hyponym *position* (e.g., "looking for permanent *job*"), others entailing the word *task* (e.g., "the *job* of repairing"). The problem of deciding whether a particular instance of *job* can be replaced by *position*, and not by the word *place*, can be solved by looking for the most similar contexts where either *position* or *place* occur in the training data, and then selecting the class (i.e., the entailed word) characterized by the most similar ones, in an instance based style. In the first example (see row 1), the word *job* is strongly associated to the word *position*, because the contexts of the latter in the examples 1 and 2 are similar to the context of the former, and not to the word *task*, whose contexts (4, 5 and 6) are radically different. On the other hand, the second example (see row 2) of the word *job* is similar to the occurrences 4 and 5 of the word *task*, allowing its correct substitution.

It is worthwhile to remark that, due to the ambiguity of the entailed words (e.g., *position* could also entail either *perspective* or *place*), not every occurrence of them should be taken into account, in order to avoid misleading predictions caused by the irrelevant senses. Therefore, approaches based on a more classical contextual similarity technique (Lin, 1998; Dagan, 2000), where words are described "globally" by context vectors, are doomed to fail. We will provide empirical evidence of this in the evaluation section.

Choosing an appropriate similarity function for the contexts of the words to be substituted is a primary issue. In this work, we exploited similarity functions already defined in the WSD literature, relying on the analogy between the lexical entailment and the WSD task. The state-of-the-art supervised WSD methodology, reporting the best results in most of the Senseval-3 lexical sample tasks in different languages, is based on a combination of syntagmatic and domain kernels (Gliozzo et al., 2005) in a SVM classification framework. Therefore, we adopted exactly the same strategy for our purposes.

A great advantage of this methodology is that it is totally corpus based, as it does not require neither the availability of lexical databases, nor the use of complex preprocessing steps such as parsing or anaphora resolution, allowing us to apply it on different languages and domains once large corpora are available for training. Therefore, we exploited exactly the same strategy to implement the IBLE classifier required for our purposes, defining a kernel composed by n simple kernels, each representing a different aspect to be considered when estimating contextual similarity among word occurrences. In fact, by using the closure properties of the kernel functions, it is possible to define the kernel combination schema as follows[3]:

$$K_C(x_i, x_j) = \sum_{l=1}^{n} \frac{K_l(x_i, x_j)}{\sqrt{K_l(x_j, x_j) K_l(x_i, x_i)}}, \quad (1)$$

where K_l are valid kernel functions, measuring similarity between the objects x_i and x_j from different perspectives[4].

One means to satisfy both the WSD and the lexical entailment requirements is to consider two different aspects of similarity: domain aspects, mainly related to the topic (i.e., the global context) of the texts in which the word occurs, and syntagmatic aspects, concerning the lexico-syntactic pattern in the local context. Domain aspects are captured by the domain kernel, described in Section 3.1, while syntagmatic aspects are taken into account by the syntagmatic kernel, presented in Section 3.2.

[3]Some recent works (Zhao and Grishman, 2005; Gliozzo et al., 2005) empirically demostrate the effectiveness of combining kernels in this way, showing that the combined kernel always improves the performance of the individual ones. In addition, this formulation allows evaluating the individual contribution of each information source.

[4]An exhaustive discussion about kernel methods for NLP can be found in (Shawe-Taylor and Cristianini, 2004).

Entailed	*job*		*Training*
position	... looking for permanent academic **job** in ...	1	... from entry-level through permanent **positions**.
		2	My academic **position** ...
		3	... put the lamp in the left **position** ...
task	The **job** of repairing	4	The **task** of setting up ...
		5	Repairing the engine is an hard **task**.
		6	... **task** based evaluation.

Table 1: IBLE example.

3.1 The Domain Kernel

(Magnini et al., 2002) claim that knowing the domain of the text in which the word is located is a crucial information for WSD. For example the (domain) polysemy among the `Computer_Science` and the `Medicine` senses of the word `virus` can be solved by simply considering the domain of the context in which it is located. Domain aspects are also crucial in recognizing lexical entailment. For example, the term `virus` entails `software_agent` in the `Computer_Science` domain (e.g., "The laptop has been infected by a *virus*"), while it entails *bacterium* when located in the `Medicine` domain (e.g., "HIV is a *virus*"). As argued in (Magnini et al., 2002), domain aspects can be considered by analyzing the lexicon in a large context of the word to be disambiguated, regardless of the actual word order. We refer to (Gliozzo et al., 2005) for a detailed description of the domain kernel. The simplest methodology to estimate the domain similarity among two texts is to represent them by means of vectors in the Vector Space Model (VSM), and to exploit the cosine similarity. The VSM is a k-dimensional space $\mathbb{R}^k$, in which the text t_j is represented by means of the vector $\vec{t_j}$ such that the i^{th} component of $\vec{t_j}$ is the term frequency of the term w_i in it. The similarity between two texts in the VSM is estimated by computing the cosine between them, providing the kernel function K_{VSM} that can be used as a basic tool to estimate domain similarity between texts[5].

[5]In (Gliozzo et al., 2005), in addition to the standard VSM, a domain kernel, exploiting external information acquired from unlabeled data, has been also used to reduce the amount of (labeled) training data. Here, given that our approach is fully unsupervised, i.e., we can obtain as many examples as we need, we do not use the domain kernel.

3.2 The Syntagmatic Kernel

Syntagmatic aspects are probably the most important evidence for recognizing lexical entailment. In general, the strategy adopted to model syntagmatic relations in WSD is to provide bigrams and trigrams of collocated words as features to describe local contexts (Yarowsky, 1994). The main drawback of this approach is that non contiguous or shifted collocations cannot be identified, decreasing the generalization power of the learning algorithm. For example, suppose that the word *job* has to be disambiguated into the sentence "…permanent academic *job* in…", and that the occurrence "We offer permanent *positions*…" is provided for training. A traditional feature mapping would extract the context words w_{-1}:`academic`, w_{-2}:`permanent` to represent the former, and w_{-1}:`permanent`, w_{-2}:`offer` to index the latter. Evidently such features will not match, leading the algorithm to a misclassification.

The syntagmatic kernel, proposed by Gliozzo et al. (2005), is an attempt to solve this problem. It is based on a gap-weighted subsequences kernel (Shawe-Taylor and Cristianini, 2004). In the spirit of kernel methods, this kernel is able to compare sequences directly in the input space, avoiding any explicit feature mapping. To perform this operation, it counts how many times a (non-contiguous) subsequence of symbols u of length n occurs in the input string s, and penalizes non-contiguous occurrences according to the number of the contained gaps. To define our syntagmatic kernel, we adapted the generic definition of the sequence kernels to the problem of recognizing collocations in local word contexts. We refer to (Giuliano et al., 2006) for a detailed description of the syntagmatic kernel.

4 Lexical Entailment for Ontology Population

In this section, we apply the IBLE technique, described in Section 3, to recognize lexical entailment for ontology population. To this aim, we cast ontology population as a lexical entailment task, where the fine-grained categories are the candidate entailed words, and the named entities to be subcategorized are the entailing words. Below, we present the main steps of our algorithm in details.

Step 1 By using a state-of-the-art supervised NER system, we recognize the named entities belonging to a set of coarse-grained categories (e.g., location and people) of interest for the domain.

Step 2 For all fine-grained categories belonging to the same coarse-grained type, we extract from a domain corpus all the occurrences of their lexicalizations in context (e.g., for the category `actor`, we extract all contexts where the term *actor* occurs), and use them as input to train the IBLE classifier. In this way, we obtain a multi-class classifier for each ontological type. Then, we classify all the occurrences of the named entities recognized in the first step. The output of this process is a list of tagged named entities; where the elements of the list could have been classified into different fine-grained categories even though they refer to the same phrase (e.g., the occurrences of the entity "Jack London" could have been classified both as `writer` and `actor`, depending on the contexts where they occur).

Step 3 A distinct category is finally assigned to the entities referring to the same phrase in the list. This is done on the basis of the tags that have been assigned to all its occurrences during the previous step. To this purpose, we implemented a voting mechanism. The basic idea is that an entity belongs to a specific category if its occurrences entail a particular superclass "more often than expected by chance", where the expectation is modeled on the basis of the overall distribution of fine-grained category labels, assigned during the second step, in the corpus. This intuition is formalized by applying a statistical reliability measure, that depends on the distribution of positive assignments for each class, defined by the following formula:

$$R(e, c) = \frac{P(c|e) - \mu_c}{\sigma_c}, \qquad (2)$$

where $P(c|e)$ is estimated by the relative frequency of the fine-grained class c among the different occurrences of the entity e, μ_c and σ_c measure the mean and the standard deviation of the distribution $P(c|E)$, and E is an (unlabeled) training set of instances of the coarse-grained type classified by the IBLE algorithm. Finally, each entity is assigned to the category c^* such that

$$c^* = \underset{c}{argmax}\ R(e, c). \qquad (3)$$

5 Evaluation

Evaluating a lexical entailment algorithm in itself is rather complex. Therefore, we performed a task driven evaluation of our system, measuring its usefulness in an ontology population task, for which evaluation benchmarks are available, allowing us to compare our technique to existing state-of-the-art approaches.

As introduced in Section 4, the ontology population task can be modeled as a lexical entailment problem, in which the fine-grained classes are the entailed words and the named entities belonging to the coarse-grained ontological type are the entailing words.

In the following, we first introduce the experimental settings (Section 5.1). Then we evaluate our technique by comparing it to state-of-the-art unsupervised approaches for ontology population (Section 5.2).

5.1 Experimental Settings

For all experiments, we adopted the evaluation benchmark proposed in (Tanev and Magnini, 2006). It considers two high-level named entity categories both having five fine-grained sub-classes (i.e., `mountain`, `lake`, `river`, `city`, and `country` as subtypes of LOCATION; `statesman`, `writer`, `athlete`, `actor`, and `inventor` are subtypes of PERSON). The authors used WordNet and Wikipedia as primary data sources for populating the evaluation ontology. In total, the ontology is populated with 280 instances which were not ambiguous (with respect to the ontology) and appeared at least twice in the English CLEF corpus[6]. Even the evaluation task is rather small and can be perceived as an artificial experimental setting, it is the best available benchmark we can use to compare our system to existing approaches in the literature, as we are not aware of other available resources.

To perform NER we used CRFs (Lafferty et al., 2001). We trained a first-order CRF on the MUC data set to annotate locations and people. In our experiments, we used the implementation provided in MALLET (McCallum, 2002). We used a standard feature set inspired by the literature on text chunking and NER (Tjong Kim Sang and Buchholz, 2000; Tjong Kim Sang and De Meulder, 2003; Tjong Kim Sang, 2002) to train a first-order CRFs. Each instance is represented by encoding all the following families of features, all time-shifted by -2,-1,0,1,2: (a) the word itself, (b) the PoS tag of the token, (c) orthographic predicates, such as *capitalization*, *upper-case*, *numeric*, *single character*, and *punctuation*, (d) gazetteers of locations, people names and organizations, (e) character-n-gram predicates for $2 \leqslant n \leqslant 3$.

As an (unsupervised) training set for the fine-grained categories, we exploited all occurrences in context of their corresponding terms we found in the CLEF corpus (e.g., for the category `actor` we used all the occurrences of the term *actor*). We did not use any prior estimation of the class frequency, adopting a pure unsupervised approach. Table 2 lists the fine-grained concepts and the number of the training examples found for each of them in the CLEF corpus.

As a reference for a comparison of the outcomes of this study, we used the results presented in (Tanev and Magnini, 2006) for the Class-Word and Class-Example approaches. The Class-Word approach exploits a similarity metric between terms and concepts based on the comparison of the contexts where they appear. Details of this technique can be found in (Cimiano and Völker, 2005). Tanev and Magnini (2006) proposed a variant of the Class-Word algorithm, called Class-Example, that relies on syntactic features extracted from corpus and uses as an additional input a set of training examples for each class. Overall, it required $1,194$ examples to accomplish this task.

[6]http://www.clef-campaign.org

All experiments were performed using the SVM package *LIBSVM*[7] customized to embed our own kernel. In all the experiments, we used the default parameter setting.

location		**person**	
mountain	1681	statesman	119
lake	730	writer	3436
river	1411	athlete	642
city	35000	actor	2356
country	15037	inventor	105

Table 2: Number of training examples for each class.

5.2 Results

Table 4 shows our results compared with two baselines (i.e., random and most frequent, estimated from the test data) and the two alternative approaches for ontology population described in the previous section. Our system outperforms both baselines and largely surpasses the Class-Word unsupervised method.

It is worthwhile to remark here that, being the IBLE algorithm fully unsupervised, improving the most frequent baseline is an excellent result, rarely achieved in the literature on unsupervised methods for WSD (McCarthy et al., 2004). In addition, our system is also competitive when compared to supervised approaches, being it only 5 points lower than the Class-Example method, while it does not require seed examples and syntactic parsing. This characteristic makes our system flexible and adaptable to different languages and domains.

System	**Micro F1**	**Macro F1**
RND Baseline	0.20	0.20
Class-Word	0.42	0.33
MF baseline	0.52	NA
IBLE	0.57	0.47
Class-Example	0.62	0.68

Table 3: Comparison of different ontology population techniques.

[7]http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Finally, we performed a disaggregated evaluation of our system, assessing the performance for different ontological types and different concepts. Results show that our method performs better on larger fine-grained classes (i.e., `writer` and `country`), while the results on smaller categories are affected by low recall, even if the predictions provided by the system tends to be highly accurate. Taking into consideration that our system is fully unsupervised, this behavior is highly desirable because it implies that it is somehow able to identify the predominant class. In addition the high precision on the smaller classes can be explained by our instance based approach.

Person	N	Prec	Rec	F1
Inventor	11	1	0.18	0.31
Statesman	20	1.0	0.05	0.10
Writer	88	0.61	0.89	0.72
Actor	25	0.57	0.68	0.62
Athlete	20	1	0.1	0.18
Micro	164	0.61	0.61	0.61
Macro	5	0.83	0.38	0.52

Table 4: Performance of the IBLE approach on people.

Location	N	Prec	Rec	F1
City	23	0.35	0.26	0.30
Country	40	0.61	0.70	0.65
River	10	0.8	0.4	0.53
Mountain	5	0.25	0.2	0.22
Lake	4	0.2	0.5	0.29
Micro	82	0.50	0.50	0.50
Macro	5	0.44	0.41	0.42

Table 5: Performance of the IBLE approach on locations.

6 Conclusions and Future Work

In this paper, we presented a novel unsupervised technique for recognizing lexical entailment in texts, namely instance based lexical entailment, and we exploited it to approach an ontology population task. The basic assumption is that if a word is entailed by another in a given context, then some of the contexts of the entailed word should be similar to that of the word to be disambiguated. Our technique is effective, as it largely surpasses both the random and most frequent baselines. In addition, it improves over the state-of-the-art for unsupervised approaches, achieving performances close to the supervised rivaling techniques requiring hundreds of examples for each class.

Ontology population is only one of the possible applications of lexical entailment. For the future, we plan to apply our instance based approach to a wide variety of tasks, e.g., lexical substitution, word sense disambiguation and information retrieval. In addition, we plan to exploit our lexical entailment as a subcomponent of a more complex system to recognize textual entailment. Finally, we are going to explore more elaborated kernel functions to recognize lexical entailment and more efficient learning strategies to apply our method to web-size corpora.

Acknowledgments

The authors would like to thank Bernardo Magnini and Hristo Tanev for providing the benchmark, Ido Dagan for useful discussions and comments regarding the connections between lexical entailment and ontology population, and Alberto Lavelli for his thorough review. Claudio Giuliano is supported by the X-Media project (`http://www.x-media-project.org`), sponsored by the European Commission as part of the Information Society Technologies (IST) program under EC grant number IST-FP6-026978. Alfio Gliozzo is supported by the FIRB-Israel research project N. RBIN045PXH.

References

Philipp Cimiano and Johanna Völker. 2005. Towards large-scale, open-domain and ontology-based named entity classification. In *Proceedings of RANLP'05*, pages 66– 166–172, Borovets, Bulgaria.

I. Dagan and O. Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. In *Proceedings of the PASCAL Workshop on Learning Methods for Text Understanding and Mining*, Grenoble.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Ido Dagan, Oren Glickman, Alfio Gliozzo, Efrat Marmorshtein, and Carlo Strapparava. 2006. Direct word sense matching for lexical substitution. In *Proceedings ACL-2006*, pages 449–456, Sydney, Australia, July.

I. Dagan. 2000. Contextual word similarity. In Rob Dale, Hermann Moisl, and Harold Somers, editors, *Handbook of Natural Language Processing*, chapter 19, pages 459–476. Marcel Dekker Inc.

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

Michael Fleischman and Eduard Hovy. 2002. Fine grained classification of named entities. In *Proceedings of ACL-2002*, pages 1–7, Morristown, NJ, USA.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1992. Work on statistical methods for word sense disambiguation. In R. Goldman et al., editor, *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 54–60.

Claudio Giuliano, Alfio Massimiliano Gliozzo, and Carlo Strapparava. 2006. Syntagmatic kernels: a word sense disambiguation case study. In *Proceedings of the EACL-2006 Workshop on Learning Structured Information in Natural Language Applications*, Trento, Italy, 5-7 April.

O. Glickman, E. Shnarch, and I. Dagan. 2006a. Lexical reference: a semantic matching subtask. In *proceedings of EMNLP 2006*.

Oren Glickman, Ido Dagan, Mikaela Keller, Samy Bengio, and Walter Daelemans. 2006b. Investigating lexical substitution scoring for subtitle generation. In *Proceedings of CoNLL-2006*.

A. Gliozzo, C. Giuliano, and C. Strapparava. 2005. Domain kernels for word sense disambiguation. In *Proceedings of ACL-2005*, pages 403–410, Ann Arbor, Michigan, June.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-2002*, pages 282–289, Williams College, MA. Morgan Kaufmann, San Francisco, CA.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of ACL-98*, pages 768–774, Morristown, NJ, USA.

B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(4):359–373.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of ACL-2004*, Barcelona, Spain, July.

J. Shawe-Taylor and N. Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Hristo Tanev and Bernardo Magnini. 2006. Weakly supervised approaches for ontology population. In *Proceedings of EACL-2006*, Trento, Italy.

Erik Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000*, Lisbon, Portugal.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147, Edmonton, Canada.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158, Taipei, Taiwan.

D. Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of ACL-94*, pages 88–95, Las Cruces, New Mexico.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of ACL 2005*, Ann Arbor, Michigan, June.