Towards large-scale, open-domain and ontology-based named entity classification[1]

Philipp Cimiano and **Johanna Völker**
Institute of Applied Informatics and Formal Description Methods
University of Karlsruhe

Abstract

Named entity recognition and classification research has so far mainly focused on supervised techniques and has typically considered only small sets of classes with regard to which to classify the recognized entities. In this paper we address the classification of named entities with regard to large sets of classes which are specified by a given ontology. Our approach is unsupervised as it relies on no labeled training data and is open-domain as the ontology can simply be exchanged. The approach is based on Harris' distributional hypothesis and, based on the vector-space model, it assigns a named entity to the contextually most similar concept from the ontology. The main contribution of the paper is a systematic analysis of the impact of varying certain parameters on such a context-based approach exploiting similarities in vector space for the disambiguation of named entities.

1 Introduction and Related Work

Named Entity Recognition (NER) systems have typically considered only a limited number of classes. The MUC named entity task (Hirschman & Chinchor 97), for example, distinguishes three classes: PERSON, LOCATION and ORGANIZATION, and the CoNLL[1] task adds one more: MISC, while the ACE framework[2] adds two more: GPE (geo-political entity) and FACILITY. Further, it has often been shown that it is relatively easy to recognize the PERSON and ORGANIZATION classes due to certain regularities, which renders MUC-like named entity recognition tasks even easier.

In this paper we propose a more challenging task, i.e. the classification of named entities with regard to a large number of classes specified by an ontology or more specifically by a concept hierarchy. Our approach aims at being open-domain in the sense that the underlying ontology and the corpus can be replaced. In our view this aim can only be accomplished if one resorts to an unsupervised system since providing labeled training data for a few hundred concepts as we consider in our approach is often unfeasible. Some researchers have addressed this challenge and have considered a larger number of classes. (Fleischman & Hovy 02) for example have considered 8 classes: ATHLETE, POLITICIAN/GOVERNMENT, CLERGY, BUSINESSPERSON, ENTERTAINER/ARTIST, LAWYER, DOCTOR/SCIENTIST and POLICE. (Evans 03) considers a totally unsupervised scenario in which the classes themselves are derived from the documents. (Hahn & Schnattinger 98) consider an ontology with 325 concepts and (Alfonseca & Manandhar 02) consider 1200 WordNet synsets. In our approach we consider an ontology consisting of 682 concepts.

Named entity recognition and classification has been so far mainly concerned with supervised techniques, the obvious drawback here being that one has to provide labeled training data for each domain and set of classes (compare (Sekine *et al.* 98; Borthwick *et al.* 98; Bikel *et al.* 99; Zhou & Su 02; G. Pailouras & Spyropoulos 00; Isozaki & Kazawa 02; Chieu & Ng 03; Hendrickx & van denBosch 03)). However, when considering hundreds of concepts as possible tags, a supervised approach requiring thousands of training examples seems quite unfeasible. On the other hand, the use of handcrafted resources such as gazetteers or pattern libraries (compare (Maynard *et al.* 03)) will also not help as creating and maintaining such resources for hundreds of concepts is equally unfeasible. Interesting and very promising are approaches which operate in a bootstrapping-like fashion, using a set of seeds to derive more training data such as the supervised approach using Hidden Markov Models in (Niu *et al.* 03) or the unsupervised approach in (Collins & Singer 99).

In this paper we present an unsupervised approach which - as many others - is based on Harris' distributional hypothesis, i.e. that words are semantically similar to the extent to which they share syntactic contexts. There have been many approaches in NLP exploiting this hypothesis, the most influential probably being the work of (Grefenstette 94) on automatic thesaurus construction as well as of (Pereira *et al.* 93) on building hierarchical clusters of nouns, the work of (Hindle 90) on discovering groups of (semantically) similar nouns as well as the work of (Yarowsky 95) and (Schuetze 98) on Word Sense disambiguation/discrimination. In particular some researchers have considered using syntactic collocations for named entity recognition (cf. (Cucchiarelli & Velardi 01) and (Lin 98)). More recently, several researchers have addressed the problem of classifying a new term into an existing ontology (Agirre *et al.* 00; Pekar & Staab 02; Alfonseca & Manandhar 02; Widdows).

In this paper we investigate the impact of using different feature weighting measures and various similarity measures described in (Lee 99). Further, to address data sparseness problems we examine the influence of (i)

[1]This is a slighlty modified version of the paper published in the proceedings of RANLP 2005

[1]http://cnts.uia.ac.be/conll2003/ner/

[2]http://www.itl.nist.gov/iaui/894.01/tests/ace/phase1/index.htm

anaphora resolution in the hope that it will yield more context information as speculated in (Grefenstette 94) (ii) downloading additional textual material from the Web as in (Agirre *et al.* 00) and making use of the structure of the concept hierarchy or taxonomy in calculating the context vectors for the classes as in (Resnik 93), (Hearst & Schütze 93) or (Pekar & Staab 02). The paper is organized as follows: first, we present our data set in Section 2 and describe our evaluation measures as well as present a few baselines for the task showing its complexity in Section 3. In section 4 we analyze the impact of varying the above mentioned parameters step by step starting with a window-based approach as a baseline. Before concluding we also discuss the results of our approach with respect to other systems performing a similar task.

2 Data Set

Our data set consists of 1880 texts containing destination descriptions downloaded from *http://www.lonelyplanet.com/destinations*. In order to create an evaluation standard, we asked two test persons to annotate the named entities of 30 randomly selected texts with the appropriate concept from a given ontology. They used a pruned version of a tourism ontology developed within an information retrieval project at our site. The original ontology consisted of 1043 concepts, but we removed some irrelevant concepts beforehand in order to facilitate the task for the annotators, resulting in an ontology with 682 concepts. In what follows, we will refer to these annotators as A and B. Annotator A actually produced 436 annotations and subject B produced 392. There were 277 named entities that were annotated by both subjects. For these 277 named entities, they used 59 different concepts and coincided in 176 cases, the agreement thus being 63.54%. The categorial agreement on these 277 named entities measured by the Kappa statistic was 63.48% (cf. (Carletta 96)), which allows to conclude that the annotation task is overall more or less well defined but that the agreement between humans is far from perfect. A system selecting a concept for a given named entity at random would thus be correct in 0.15% cases, which already shows the difficulty of the task. We evaluate our system on the named entities annotated by both subjects as described in the following section. It is important to emphasize however that we totally abstract here from the actual recognition of named entities in the sense that the input to our system is a set of named entities to be assigned to the appropriate class.

3 Evaluation

As mentioned in (Collins & Singer 99), the task in named entity recognition is to learn a function from an input string (a proper name) to its class. In particular our aim is to learn a function f_S which approximates the functions f_A and f_B specified by both annotators. We assume that these functions are given as sets $C_X := \{(e,c) | e \in dom(f_X) \wedge f_X(e) = c\}$, where e is the named entity in question, c is the concept it has been assigned to and $dom(f)$ is the domain of a function f. While f_A and f_B are total functions, f_S is a partial one as our system does not always produce an answer. In fact, if the distributional similarity between the entity to be tagged and all the concepts in the ontology is minimal, then the system will give no answer. Thus it is not only important to measure the recall, but also the precision of the system. We thus evaluate the system with the standard measures of Precision, Recall and F-Measure, i.e.

$$P_A = \frac{|C_A \cap C_S|}{|C_S|} \quad P_B = \frac{|C_B \cap C_S|}{|C_S|} \quad P = \frac{P_A + P_B}{2}$$
$$R_A = \frac{|C_A \cap C_S|}{|C_A|} \quad R_B = \frac{|C_B \cap C_S|}{|C_B|} \quad R = \frac{R_A + R_B}{2}$$
$$F_A = \frac{2 * P_A * R_A}{P_A + R_A} \quad F_B = \frac{2 * P_B * R_B}{P_B + R_B} \quad F = \frac{F_A + F_B}{2}$$

As named entities can be tagged at different levels of detail and there is certainly not only one correct assignment of a concept, we also consider how close the assignment of the system is with respect to the assignment of the annotator by using the *Learning Accuracy* originally introduced by (Hahn & Schnattinger 98). However, we consider a slightly different and symmetric formulation of the Learning Accuracy in line with the measures defined in (Maedche *et al.* 02). The main difference is that we measure the distance between nodes in terms of edges – in contrast to nodes in Hahn's version – and we do not need any case distinction taking into account if the classification was correct or not. The Learning Accuracy is defined as follows:

$$LA(a,b) := \frac{\delta(top,c) + 1}{\delta(top,c) + \delta(a,c) + \delta(b,c) + 1}$$

where $c = lcs(a,b)$ is the least common subsumer of concepts a and b as defined in (Maedche *et al.* 02).

4 Experiments

As mentioned above, our approach is in line with Harris' distributional hypothesis and other approaches in which the context of a phrase is used to disambiguate its sense (Yarowsky 95; Schuetze 98) or class (Lin 98) or to discover other semantically related terms (Hindle 90). As other approaches, we also adopt the one-sense-per-discourse assumption (Gale *et al.* 92), i.e. we do not perform any word sense disambiguation. Our algorithm thus assigns an instance represented by a certain context vector $\vec{i}$ to the concept corresponding to the most similar vector $\vec{c}$. The algorithm is basically as follows:

```
classify(set of instances I, corpus t, set of concepts C){
    foreach c in C
        v_c = getContextVector(c,t);
    foreach foreach c in C
        doFeatureWeighting(v_c);
    foreach i in I {
        v_i = getContextVector(i,t);
        class(i)=maxarg_c sim(v_c,v_i);
    }
    return class;
}
```

Though most approaches represent the context of a phrase as a vector, there are great differences in which features are used ranging from simple word windows (Yarowsky 95; Schuetze 98) to syntactic dependencies extracted with a parser (Hindle 90; Pereira *et al.* 93; Grefenstette 94). We start our analysis by comparing window-based techniques with using pseudo-syntactic dependencies extracted by using regular expressions over part-of-speech tags. Furthermore, we analyze the impact of using different similarity and feature weighting measures. As they were found to perform particularly well in (Lee 99), we use the following similarity measures: the *cosine* and *Jaccard* measures, the *L1 norm* as well as the *Jensen-Shannon* and the *Skew* divergence. Further, we weight the features according to different measures. In particular, we use the following measures:

$$Conditional(n, feat) = P(n|feat) = \frac{f(n,feat)}{f(feat)}$$

$$PMI(n, feat) = log\,\frac{P(n|feat)}{P(n)}$$

$$Resnik(n, feat) = S_R(feat)\,P(n|feat)$$

where $S_R(feat) = \sum_{n'} P(n'|feat)\,log\frac{P(n'|feat)}{P(n')}$.

Furthermore, $f(n, feat)$ is the number of occurrences of a term n with feature $feat$, $f(feat)$ is the number of occurrences of the feature $feat$ and $P(n)$ is the relative frequency of a term n compared to all other terms. The first information measure is simply the conditional probability of the term n given the feature $feat$. The second measure $PMI(n, v)$ is the so called *pointwise mutual information* and was used by (Hindle 90) for discovering groups of similar terms. The third measure is inspired by the work of (Resnik 93) and introduces an additional factor $S_R(n, feat)$ which takes into account all the terms appearing with the feature in question. In particular, the factor measures the relative entropy of the prior and posterior (given the feature) distributions of n and thus the 'selectional strength' of a given feature.

4.1 Using Word Windows

In a first experiment we used the n words to the left and right of a certain word of interest excluding so called stopwords and without trespassing sentence boundaries. Here n is the so called window size. The advantage of such an approach is that no preprocessing is necessary to extract context information. However, it also has the drawback of making context vectors larger than when using syntactic dependencies thus making the similarity calculation less efficient (cf.(Grefenstette 94)). We implemented this approach in order to verify if syntactic dependencies actually perform better in our setting. We varied the similarity measure, the feature weighting strategy as well as experimented with the three different window sizes: 3, 5 and 10 words. thus producing 5 * 4 * 3 = 60 runs of the similarity-based classification algorithm. Due to space limitations we do not present all the results. The best result was an F-Measure of 19.7% and a Learning Accuracy of 57.78%. It was achieved when using the Skew divergence as similarity measure, *Resnik* as feature weighting measure and a window size of 10.

4.2 Using pseudo-syntactic dependencies

Instead of merely using the words occurring within a given window size before and after the word in question, we also experimented with using pseudo-syntactic dependencies. These dependencies are not really syntactical as they are not obtained from parse trees, but from a very shallow method consisting in matching certain regular expression over part of speech tags. The motivation for doing this is the observation in (Grefenstette 94) that the quality of using word windows or syntactic dependencies for distributional analysis depends on the rank or frequency of the word in question. In this line, our intention is to make a compromise between using word windows and syntactic dependencies extracted from parse trees. Our pseudo-syntactic dependencies are surface dependencies extracted by matching regular expressions. In what follows we list the syntactic expressions we use and give a brief example of how the features, represented as predicates, are extracted from these expressions:

- adjective modifiers, i.e. *a nice city* → nice(city)
- prepositional phrase modifiers, i.e. *a city near the river* → near_river(city) and city_near(river), respectively
- possessive modifiers, i.e. *the city's center* → has_center(city)
- noun phrases in subject or object position. i.e. *the city offers an exciting nightlife* → offer_subj (city) and offer_obj(nightlife)
- prepositional phrases following a verb, i.e. *the river flows through the city* → flows_through(city)
- copula constructs i.e. *a flamingo is a bird* → is_bird(flamingo)
- verb phrases with the verb *to have*, i.e. *every country has a capital* → has_capital(country)

Consider for example the following discourse:

Mopti is the biggest city along the Niger with one of the most vibrant ports and a large bustling market. Mopti has a traditional ambience that other towns seem to have lost. It is also the center of the local tourist industry and suffers from hard-sell overload. The nearby junction towns of Gao and San offer nice views over the Niger's delta.

Here we would extract the following concept vectors:

city: biggest(1)
ambience: traditional(1)

center: of_tourist_industry(1)
junction towns: nearby(1)
market: bustling(1)
port: vibrant(1)
tourist industry: center_of(1), local(1)
town: seem_subj(1)
view: nice(1), offer_obj(1)

and the following ones for named entities:

Mopti: is_city(1), has_ambience(1)
San: offer_subj(1)
Gao: junction_of(1)
Niger: has_delta(1)

Table 1 shows the results for the version of the classification algorithm making use of the pseudo-syntactic dependencies using the different similarity and feature weighting measures (Standard). The best result was an F-Measure of 19.58% and a Learning Accuracy of 60.03%. The fact that the F-Measure is slightly worse is definitely compensated by a higher Learning Accuracy. Furthermore, as the length of the vectors is much smaller and thus the computation of the similarities faster, we conclude that using the pseudo-syntactic dependencies is an interesting alternative and present the results of further modifications to our algorithm with respect to the version using these sort of dependencies.

4.3 Dealing with Data Sparseness

4.3.1 Using Conjunctions

In order to address the problem of data sparseness we exploit conjunctions of named entities in the sense that if two named entities appear linked by the conjunctions 'and' or 'or', we count any occurrence of a feature with one of the named entities also as an occurrence of the other. As the results in Table 1 show, this simple heuristic improves the results of our approach considerably. The top results are F-Measures of 22.8% (Cosine), 22.57% (L1 norm) and 22.57% (Skew divergence) with a Learning Accuracy of 61.23%, 61.4% and 62.7%, respectively.

4.3.2 Exploiting the Taxonomy

An interesting option discussed in (Resnik 93), (Pekar & Staab 02) and (Hearst & Schütze 93) is to take into account the taxonomy of the underlying ontology to compute the context vector of a certain term by taking into account the context vectors of its hyponyms. This is in fact a delicate issue as some studies have shown that this doesn't work while other have shown the contrary. We adopt here a conservative strategy and take only into account the context vectors of direct hyponyms to compute the vector of a certain term. In fact, the context vector of a hypernym will be the sum of the context vectors of all its direct hyponyms. We assume a one-to-one mapping between nouns and concept labels, thus considering the hyponyms of all possible concepts for a given label. We will refer to this as the 'standard' version. However, the aggregated vectors can also be normalized. In fact, we experiment with the two possibilities also discussed in (Pekar & Staab 02): (i) standard normalization of the vector or (ii) calculating its centroid (compare (Pekar & Staab 02) and (Hearst & Schütze 93)). In the latter the only difference is that we create an average vector by dividing through the number of direct hyponyms. As the results in Table 1 show, only the version with the centroid method did indeed yield better results, while the standard (no vector normalization) and the category method (standard vector normalization) did actually make the results worse. The best result with the centroid method was an F-Measure of 23.02% and a Learning Accuracy of 64.11%.

4.3.3 Anaphora Resolution

As another approach to overcome the problem of data sparseness we explored the impact of anaphora resolution on the task of named entity recognition. Based on MINIPAR (cf. (Lin 93)) and the work by (Lappin & Leass 94) we implemented an algorithm for identifying intrasentential antecedents of 3rd person personal and possessive pronouns which replaces each (non-pleonastic) anaphor by the grammatically correct form of the corresponding antecedent as shown in the following examples:

The port capital of Vathy is dominated by ***its*** *fortified Venetian harbour.* →
The port capital of Vathy is dominated by ***Vathy's*** *fortified Venetian harbour.*
Holiday hooligans used to head to nearby Benitses, until ***it*** *was ruined, so now* ***they*** *head north to cut a swathe through the coastline's few remaining unspoilt coves and fishing villages.*→
Holiday hooligans used to head to nearby Benitses, until ***Benitses*** *was ruined, so now* ***the hooligans*** *head north to cut a swathe through the coastline's few remaining unspoilt coves and fishing villages.*

Moreover, in order to improve the detection of pleonastic occurrences of *it*, we use a modified set of patterns developed by (Dimitrov 02). Although our implementation seems to perform a bit worse than the one by Lappin and Leass (maybe due to the very noisy data set) the evaluation yielded a remarkable precision of about 0.79 and a recall of approximately 0.7.

As shown by Table 1 the use of anaphora resolution even improves the results we obtained by exploiting the taxonomy leading to an F-Measure of 23.82% and a Learning Accuracy of 65.04% (Skew divergence).

4.3.4 Downloading Documents from the Web

Since named entities tend to occur less often than common nouns representing possible classes, they are to a particularly high degree affected by the problem of data sparseness. We address this issue by downloading from the web a set of at most 20 additional documents D_i for each named entity i. Moreover, in order make sure that each $d \in D_i$ belongs to the correct sense of i we compare d with all documents in the original corpus containing at least one occurrence of i. The decision whether to keep d or not is made by creating bag-of-words style vectors rep-

	Cosine		Jaccard		L1		JS		Skew	
	F	LA	F	LA	F	LA	F	LA	F	LA
Standard										
Frequency	13.29%	55.77%	1.4%	29.99%	15.62%	59.45%	2.56%	39%	14.45%	59.41%
Conditional	16.78%	58.47%	1.4%	29.99%	18.65%	59.31%	6.29%	41.86%	17.02%	58.71%
PMI	19.11%	58.93%	1.4%	29.99%	17.72%	57.29%	5.13%	40.25%	**19.58%**	**60.03%**
Resnik	15.38%	56.33%	1.4%	29.99%	18.18%	58.91%	4.9%	38.12%	19.35%	60.44%
Conjunctions										
Frequency	18.51%	61.25%	11.54%	44.22%	18.28%	63.58%	10.16%	52.06%	21.9%	65.19%
Conditional	20.77%	60.87%	11.54%	44.22%	21.9%	63.27%	11.06%	43.46%	22.12%	63.41%
PMI	**22.8%**	**61.23%**	11.54%	44.37%	**22.57%**	**61.4%**	10.84%	42%	**22.57%**	**62.7%**
Resnik	21.22%	60.32%	11.54%	44.37%	22.12%	61.71%	10.61%	43.1%	22.35%	62.92%
Conjunctions + Ontology										
Freuqency	5.42%	63.12%	11.09%	44.93%	5.42%	66.82%	10.61%	51.18%	5.42%	65.82%
Conditional	5.64%	64.04%	11.09%	44.93%	5.64%	64.46%	10.84%	46.09%	5.64%	64.99%
PMI	6.32%	64.17%	11.09%	44.81%	5.87%	63.59%	10.61%	43.59%	5.87%	63.43%
Resnik	5.42%	62.52%	11.09%	44.81%	5.87%	62.78%	11.06%	44.88%	5.87%	63.39%
Conjunctions + Ontology (Category)										
Frequency	10.16%	47.84%	11.09%	44.93%	13.77%	55.78%	10.61%	51.18%	14.67%	59.79%
Conditional	3.16%	42.84%	11.09%	44.93%	5.42%	49.7%	10.84%	46.09%	6.77%	58.04%
PMI	5.87%	45.76%	11.09%	44.81%	9.71%	44.03%	1.36%	38.65%	7.9%	53.71%
Resnik	5.19%	43.16%	11.09%	44.81%	6.55%	49.14%	0.9%	34.92%	6.32%	59.06%
Conjunctions + Ontology (Centroid)										
Frequency	22.35%	63.57%	11.09%	44.93%	23.02%	63.27%	10.61%	51.18%	13.54%	62.63%
Conditional	22.12%	61.05%	11.09%	44.93%	22.8%	62.53%	10.84%	46.09%	**23.02%**	**64.11%**
PMI	22.12%	60.66%	11.09%	44.81%	22.8%	61.72%	10.38%	42.33%	19.86%	63.47%
Resnik	20.99%	60.62%	11.09%	44.81%	22.12%	61.89%	10.61%	43.39%	21.9%	64.33%
Conjunctions + Ontology (Centroid) + Anaphora Resolution										
Frequency	22.25%	64.8%	10.59%	42.8%	23.15%	65.45%	10.11%	49.12%	15.28%	65.17%
Conditional	22.7%	62.19%	10.59%	42.8%	23.37%	63.92%	11.01%	45.58%	**23.82%**	**65.04%**
PMI	22.92%	61.69%	10.59%	43.1%	23.6%	63.32%	11.24%	43.6%	18.88%	64.49%
Resnik	22.25%	61.06%	10.59%	43.1%	23.37%	63.42%	10.36%	43.16%	23.37%	64.69%
Conjunctions + Ontology (Centroid) + Web Crawling										
Frequency	25.4%	65.43%	12.1%	51.01%	24.4%	64.22%	6.25%	45.61%	9.07%	64.68%
Conditional	25.6%	64.46%	12.1%	51.01%	25.81%	64.43%	3.63%	39.72%	**26.21%**	**65.91%**
Mutual	25.6%	63.94%	10.08%	50.4%	25.81%	63.72%	3.43%	23.63%	12.1%	64.31%
Resnik	24.4%	61.9%	10.08%	50.4%	24.6%	62.41%	1.81%	20.17%	25.2%	65.18%

Table 1: Results for pseudo-syntactic dependencies

resentations for each of the involved documents, computing their cosine and only considering the document if the similarity is over an experimentally determined threshold of 0.2. Table 1 shows that this way of extending the corpus with documents from the web considerably improves all previous results. With the Skew divergence we achieved an F-Measure of 26.21% and a Learning Accuracy of 65.91%.

4.3.5 Postprocessing

Finally, we also examine a postprocessing step in which the k best answers of the system (ranked according to their corresponding similarities from highest to lowest) are checked for their statistical plausibility on the Web. For this purpose, inspired by the work of (Markert *et al.* 03), for each named entity e and the top k answers $c_1, .., c_k$ we generate the following Hearst-style (Hearst 92) pattern strings and count their occurrences on the Web by using the Google Web API:

1. $\pi(c_i)$ such as e
2. e and other $\pi(c_i)$
3. e or other $\pi(c_i)$
4. $\pi(c_i)$, especially e
5. $\pi(c_i)$, including e

where $\pi(w)$ is the result of looking up the plural form of the word w in a lexicon containing inflected forms and their corresponding lemmas. Furthermore, the number of hits of the above pattern string are normalized by dividing through the number of hits of the underlined parts. At the end, that answer of the k best is chosen which maximizes this coefficient. We experimented with different values for k, i.e. 3, 5 and 10. This extension is furthermore efficient as we only need to generate $k + 1$ queries to the Google Web API for each named entity. Table 2 gives the results of this step when postprocessing the results produced with the versions of our system using anaphora resolution and crawling documents from the Web. The results show that the F-Measures increase considerably when using our postprocessing step. The best result is an F-Measure of 32.6% with a precision of 36.82%, a recall of 29.34% and a Learning Accuracy of 69.87% for the version of our system crawling the Web.

4.3.6 Discussion

The best result of our approach is an F-Measure of 32.6% which is more than 32 points points above the naive baseline of F=0.15%, almost 20 points over the majority-class-baseline of F=12.64% and 12.9 points over the word-window-based approach approach with a window size of 10 (F=19.7%). When considering this best version of our approach, the precision is 36.82% and the recall 29.34%. In order to compare our results with systems performing a similar task, we compare our recall as well as Learning Accuracy with the one of the systems in Table 3. In fact, our recall value corresponds to the accuracy values of the other approaches. (Fleischman & Hovy 02) for example

k	k=3				k=5				k=10			
	F	P	R	LA	F	P	R	LA	F	P	R	LA
AR	29.15%	38.46%	23.47%	71.04%	28.7%	37.87%	23.1%	71%	30.72%	40.53%	24.73%	71.71%
WC	30.58%	34.54%	27.44%	67.71%	30.78%	34.77%	27.62%	68.52%	**32.6%**	36.82%	29.24%	**69.87%**

Table 2: Results of the postprocessing step on the A(naphora) R(esolution) and the W(eb) C(rawling versions))

make use of a supervised approach and extract n-grams for training several classifiers. (Evans 03) computes hypernym vectors for each entity by using the Google API and clusters instances on the basis of these, thus considering a totally unsupervised scenario in which the classes themselves are derived from the data. (Alfonseca & Manandhar 02) present a similar approach to ours relying on distributional similarity and achieve the best results using verb-object dependencies as features, while (Hahn & Schnattinger 98) present an elaborated qualification calculus for reasoning about the quality of different hypothesis. The systems thus rely on different assumptions, learning paradigms as well as number of classes, such that they are not directly comparable. The conclusions which can be drawn from Table 3 are that (i) obviously the task is the harder the more classes are considered and (ii) our approach fits very well from a quantitative point of view into the landscape of systems performing a similar – but not equivalent – task. Considering the most similar systems, it is worth mentioning that our results are worse than the ones of (Hahn & Schnattinger 98), which however consider half as many concepts and furthermore assume a perfect syntactic and semantic analysis as well as an elaborated DL concept hierarchy. On the other hand we achieve much better results than (Alfonseca & Manandhar 02), but they also consider a larger number of classes. SemTag (Dill *et al.* 03) also considers a large amount of classes from the TAP ontology, but assumes that the possible classes or tags for each instance are known in advance. Thus, the system effectively performs sense disambiguation with respect to a much smaller set of classes per instance.

System	#concepts	Rec/Acc	LA
MUC	3	>90%	n.a
Fleischman et al.	8	70.4%	n.a.
Evans	2-8	41.41%	n.a.
Hahn et al. (Baseline)	325	21%	67%
Hahn et al. (TH)	325	26%	73%
Hahn et al. (CB)	325	31%	76%
BEST	682	29.24%	69.87%
Alfonseca et al.	1200	17.39%	44%

Table 3: Comparison of results

5 Conclusion

We have addressed the problem of tagging named entities with regard to a large set of concepts as specified within a given concept hierarchy. In particular we have presented an approach relying on Harris' distributional hypothesis as well as on the vector-space model and assigning a named entity to that concept which maximizes the contextual similarity with the named entity in question. The aim has not been to present a fully fledged system performing this task, but to investigate the impact of varying a number of parameters. In this line we have shown that the pseudo-syntactic dependencies we have considered are an interesting alternative to window-based approaches as they yield a higher Learning Accuracy and also allow a more efficient computation of the similarities. To address the typical data sparseness problems one encounters when working with corpora, we have examined the impact of (i) exploiting conjunctions, (ii) factoring the underlying taxonomy into the computation of the concept vectors as in (Pekar & Staab 02), (iii) getting additional context by applying an anaphora resolution algorithm developed for this purpose and (iv) additionally downloading additional documents from the World Wide Web as in (Agirre *et al.* 00), showing that with the correct settings all these techniques improve the results of our approach both in terms of F-Measure and Learning Accuracy. Finally, we have also presented a postprocessing step by which the system's k most highly ranked answers are checked for their statistical plausibility on the Web, which notably improves the results of the approach. In general, the best results were achieved using the conditional probability as feature weighting strategy and the Skew divergence as similarity measure, thus confirming the results obtained in (Lee 99).

Acknowledgements: We would like to acknowledge support from the EU-IST project SEKT (IST-2003-506826) as well as the SmartWeb project, funded by the German Ministry for Education and Research.

References

(Agirre *et al.* 00) E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the WWW. In *Proceedings of the Workshop on Ontology Construction of the ECAI*, 2000.

(Alfonseca & Manandhar 02) E. Alfonseca and S. Manandhar. Extending a lexical ontology by a combination of distributional semantics signatures. In *Proceedings of the 13th EKAW*, 2002.

(Bikel *et al.* 99) D.M. Bikel, R.L. Schwartz, and R.M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.

(Borthwick *et al.* 98) A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Proc. of the Sixth ACL Workshop on Very Large Corpora*, 1998.

(Carletta 96) J. Carletta. Asessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

(Chieu & Ng 03) H.L. Chieu and H.T. Ng. Named entity recognition with a maximum entropy approach. In

Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003), pages 160–163, 2003.

(Collins & Singer 99) M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

(Cucchiarelli & Velardi 01) A. Cucchiarelli and P. Velardi. Unsupervised named entity recognition using syntatic and semantic contextual evidence. *Computational Linguistics*, 27(1):123–131, 2001.

(Dill *et al.* 03) S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin, and J.Y. Zien. SemTag and Seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th International Conference on the World Wide Web (WWW'03)*, pages 178–186, 2003.

(Dimitrov 02) M. Dimitrov. A light-weight approach to coreference resolution for named entities in text. Unpublished M.Sc. thesis, University of Sofia, 2002.

(Evans 03) R. Evans. A framework for named entity recognition in the open domain. In *Proceedings of RANLP*, pages 137–144, 2003.

(Fleischman & Hovy 02) M. Fleischman and E. Hovy. Fine grained classification of named entities. In *Proceedings of COLING*, 2002.

(G. Pailouras & Spyropoulos 00) V. Karkaletsis G. Pailouras and C.D. Spyropoulos. Learning decision trees for named-entity recognition and classification. In *Proceedings of the ECAI Workshop on Machine Learning for Information Extraction*, 2000.

(Gale *et al.* 92) W. Gale, K. Church, and Y. Yarowsky. One sense per discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*, pages 233–237, 1992.

(Grefenstette 94) G. Grefenstette. *Explorations in Authomatic Thesaurus Construction*. Kluwer, 1994.

(Hahn & Schnattinger 98) U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *Proceedings of AAAI'98/IAAI'98*, 1998.

(Hearst & Schütze 93) M.A. Hearst and H. Schütze. Customizing a lexicon to better suit a computational task. In *Proceedings of the ACL SIGLEX Workshop on Acquisition of Lexical Knowledge from Text*, 1993.

(Hearst 92) M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*, 1992.

(Hendrickx & van denBosch 03) I. Hendrickx and A. van den Bosch. Memory-based one-step named-entity recognition: Effects of seed list features, classifier stacking, and unannotated data. In *Proceedings of CoNLL-2003*, pages 176–179, 2003.

(Hindle 90) D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the Annual Meeting of the ACL*, pages 268–275, 1990.

(Hirschman & Chinchor 97) L. Hirschman and N. Chinchor. Muc-7 named entity task definition. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, 1997.

(Isozaki & Kazawa 02) H. Isozaki and H. Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of COLING*, 2002.

(Lappin & Leass 94) S. Lappin and H.J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.

(Lee 99) L. Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 25–32, 1999.

(Lin 93) D. Lin. Principle-based parsing without overgeneration. In *Proceedings of the Annual Meeting of the ACL*, pages 112–120, 1993.

(Lin 98) D. Lin. Using collocation statistics in information extraction. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

(Maedche *et al.* 02) A. Maedche, V. Pekar, and S. Staab. Ontology learning part one - on discovering taxonomic relations from the web. In *Web Intelligence*. Springer, 2002.

(Markert *et al.* 03) K. Markert, N. Modjeska, and M. Nissim. Using the web for nominal anaphora resolution. In *EACL Workshop on the Computational Treatment of Anaphora*, 2003.

(Maynard *et al.* 03) D. Maynard, K. Bontcheva, and H. Cunningham. Towards a semantic extraction of named entities. In *Proceedings of RANLP*, 2003.

(Niu *et al.* 03) C. Niu, W. Lei, J. Ding, and R.K. Srihari. A bootstrapping approach to named entity classification using successive learners. In *Proceedings of the 41st Annual Meeting of the ACL*, pages 335–342, 2003.

(Pekar & Staab 02) V. Pekar and S. Staab. Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. *Proceedings of COLING*, 2002.

(Pereira *et al.* 93) F. Pereira, N. Tishby, and L. Lee. Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting of the ACL*, pages 183–190, 1993.

(Resnik 93) P. Resnik. *Selection and Information: A Class-Based Approach to Lexical Relationships*. Unpublished PhD thesis, 1993.

(Schuetze 98) H. Schuetze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

(Sekine *et al.* 98) S. Sekine, R. Grishman, and H. Shinnou. A decision tree method for finding and classifying names in japanese texts. In *Proceedings of the Sixth ACL Workshop on Very Large Corpora*, 1998.

(Widdows) D. Widdows. Unsupervised method for developing taxonomies by combining syntactic and statistical information. In *Proceedings of HLT/NAACL*.

(Yarowsky 95) D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings od the Annual Meeting of the ACL*, pages 189–196, 1995.

(Zhou & Su 02) G. Zhou and J. Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Meeting of the ACL*, pages 473–480, 2002.