Gimme' The Context: Context-driven Automatic Semantic Annotation with C-PANKOW

Philipp Cimiano[1], Günter Ladwig[1], Steffen Staab[2,3]
[1]Institute AIFB, University of Karlsruhe, Germany
{pci, gula}@aifb.uni-karlsruhe.de
http://www.aifb.uni-karlsruhe.de/WBS
[2]Ontoprise GmbH, Karlsruhe, Germany
http://www.ontoprise.com/
[3]Institute for Computer Science, University of Koblenz-Landau, Germany
http://www.uni-koblenz.de/FB4/

ABSTRACT

Without the proliferation of formal semantic annotations, the Semantic Web is certainly doomed to failure. In earlier work we presented a new paradigm to avoid this: the 'Self Annotating Web', in which globally available knowledge is used to annotate resources such as web pages. In particular, we presented a concrete method instantiating this paradigm, called PANKOW (Pattern-based ANnotation through Knowledge On the Web). In PANKOW, a named entity to be annotated is put into several linguistic patterns that convey competing semantic meanings. The patterns that are matched most often on the Web indicate the meaning of the named entity — leading to automatic or semi-automatic annotation.

In this paper we present C-PANKOW (Context-driven PANKOW), which alleviates several shortcomings of PANKOW. First, by downloading abstracts and processing them off-line, we avoid the generation of large number of linguistic patterns and correspondingly large number of Google queries. Second, by linguistically analyzing and normalizing the downloaded abstracts, we increase the coverage of our pattern matching mechanism and overcome several limitations of the earlier pattern generation process. Third, we use the annotation context in order to distinguish the significance of a pattern match for the given annotation task. Our experiments show that C-PANKOW inherits all the advantages of PANKOW (no training required etc.), but in addition it is far more efficient and effective.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*; I.7.1 [**Document and Text Processing**]; I.2.7 [**Natural Language Processing**]

General Terms

Algorithms, Experimentation, Performance

Keywords

Annotation, Metadata, Information Extraction, Semantic Web

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-046-9/05/0005.

1. INTRODUCTION

The current situation of the Semantic Web is one of a vicious cycle in which there is not much of a semantic web due to the lack of annotated web pages, and there is such a lack because annotating web pages currently does not provide much benefit. Thus, the application of (semi-) automatic techniques in order to support web page annotation is a key factor for the vision of the Semantic Web to become true.

Several supervised machine learning based techniques have been proposed to automate the information extraction as well as annotation process of documents [4, 9, 20, 31]. However, these techniques rely on assumptions that are not compatible with the vision of the Semantic Web. First, machine learning approaches inducing extraction rules for each concept from training data such as in [4, 9, 31] do typically not scale to large numbers of concepts as Semantic Web ontologies consist of. Second, in order to annotate with respect to a few hundred concepts, a training set in the magnitude of thousands of examples needs to be provided[1], an effort that probably not many people are willing to make. Third, machine-learning based approaches rely on the assumption that documents have a similar structure as well as content, an assumption which seems quite unrealistic considering the heterogeneity of the current web.

Thus, several researchers have started to look at unsupervised approaches such as [15] as well as approaches performing a first unsupervised step and then using the results of this first step to induce new extraction rules in a bootstrapping manner [3, 10, 16].

Further, as a way out of the above mentioned vicious cycle, in [7] we presented our vision of a 'Self-Annotating Web' in which globally available syntactic resources are considered to support metadata creation. The main idea herein is to approximate semantics by considering information about the statistical distribution of certain syntactic structures over the Web. Our concrete instantiation of this paradigm is called PANKOW (Pattern-based ANnotation through Knowledge On the Web). The core of PANKOW was a pattern generation mechanism which creates pattern strings out of a certain pattern schema conveying a specific semantic relation, an instance to be annotated and all the concepts from a given ontology. It counts the occurrences of these pattern strings on the Web using the Google API. The ontological instance in question is then annotated semantically according to a principle of maximal evidence, i.e. with the concept having the largest number of hits. Our approach is thus unsupervised as it relies on no hand labeled train-

[1]Our experiences with the Amilcare system in [9] showed that at least ten examples for each concept to be extracted are necessary.

ing examples and does not assume that documents have a similar structure, thus avoiding two main problems with which supervised techniques are confronted.

Let's for example assume that the string 'Niger' appears in a web page and we have no idea about how to annotate it. Figure 1 shows the Google hits for the following four expressions: *Niger is a country*, *Niger is a state*, *Niger is a river* and *Niger is a region*. Intuitively, given these figures we would naturally tend to annotate *Niger* as a country as it seems to be its main meaning on the Web. This illustrates the fact that formal (semantic) annotations can be approximated to a certain extent by considering the statistical distribution of certain syntactic structures over the web. However, as Niger can be a country or a river depending on the context in which it appears, the example also shows that ambiguity is an important problem we need to deal with in such an approach. This paper presents C-PANKOW (Context-driven and Pattern-based Annotation through Knowledge on the Web), which tackles the ambiguity problem by taking into account the context the entity to be annotated appears in.

The predecessor of C-PANKOW in fact suffered from a few shortcomings. First, due to the restrictions of the pattern generation process, a lot of actual instances of the pattern schemes were not found. In particular the approach exhibited problems generating the correct plural forms of concept labels as well as matching more complex linguistic structures such as noun phrases including determiners, noun modifiers etc. We overcome this problem by actually downloading the pages, analyzing them linguistically, and matching the patterns instead of merely generating them and counting their Google hits. The results of the pattern-matching are also linguistically normalized, i.e. words are mapped to their base forms thus completely solving the problem with the generation of plural forms.

At the same time we overcome the second problem in this way, i.e. the large number of queries sent to the Google Web API. In fact, by downloading the pages and processing them locally we reduce network traffic. PANKOW issued a number of Google queries proportional to the size of the ontology considered. Thus PANKOW was difficult to scale to large ontologies. In C-PANKOW, we generate only a constant number of queries per instance that we annotate. Thus, C-PANKOW is able to annotate using very large ontologies. Though PANKOW was already able to take into account more concepts than standard named entity recognition systems, C-PANKOW thus definitely overcomes the scalability problem with which supervised techniques are faced.

Third and most important, we contextualize the pattern matching by distinguishing between relevant and non-relevant pages. A pattern matched in a relevant web page counts more than one matched in a less relevant one. Hereby, relevance assessment boils down to calculating the similarity of the involved pages. We present an evaluation of our system analyzing the impact of our notion of contextual relevance as well as varying the number of pages downloaded. Thereby, our experiments show that C-PANKOW outperforms its competitors.

The remainder of this paper is structured as follows: Section 2 describes the process of C-PANKOW. Section 3 presents an evaluation of the approach. Section 4 describes the implementation of the system as a freely accessible web service. Before concluding, we discuss some related work in section 5.

2. THE PROCESS OF C-PANKOW

The process of C-PANKOW is schematically described by the pseudocode in Figure 2 and is summarized in the following:

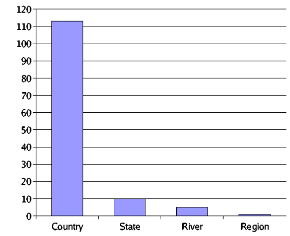

Figure 1: Statistical distribution of 'is a'-patterns for Niger

1. The web page to be annotated is scanned for candidate instances. This process is described in detail in Section 2.1.
2. Then, for each instance i discovered and for each clue/pattern-pair in our pattern library P (described in Section 2.4), we issue an automatically generated exact query to Google™ and download the abstracts of the n first hits (cf. Section 2.2).
3. We calculate the similarity between the document to be annotated and each of the downloaded abstracts. If the similarity is above a given threshold t, the actual pattern found in the abstract reveals a phrase, which may possibly describe the concept that the instance may belong to. Section 2.3 describes how the similarity is calculated.
4. Then the results for instance i are updated according to the similarity previously calculated, i.e. a pattern matched in an abstract that is very similar to the page to be annotated counts more than a pattern that is matched in a less similar abstract.
5. Finally, the instance i is annotated with that concept c having the largest number as well as most contextually relevant hits.

In what follows, we describe in detail every important step of the algorithm. The *recognizeInstances()* procedure is described in Section 2.1. Section 2.2 describes the process of downloading Google-abstracts, whereas Section 2.3 describes how the similarity is computed. Finally, Section 2.4 describes our pattern library and Section 2.5 discusses some complexity issues. The whole process is illustrated with a running example. In fact, we describe the result of each process step on the web page depicted in Figure 3.

2.1 Instance Recognition

In order to detect candidate instances in a web page we first eliminate the complete HTML markup and extract the text body of the page. This step is necessary because we apply a part-of-speech tagger to assign word categories to every token of the extracted text and the tagger is not able to handle HTML markup.[2] Then we split the text into sentences and interpret as an instance every string which matches the following pattern:

INSTANCE := (\w+{DT})? ([a-z]+{JJ})? PRE (MID POST)?

PRE := POST := (([A-Z][a-z]*){NNS|NNP|NN|NP|JJ|UH})+

MID := the{DT} | of{IN} | −{−} | ′{POS} |

[2] We use the QTag part-of-speech tagger in http://web.bham.ac.uk/O.Mason/software/tagger/. QTag's part-of-speech tagset can be found at http://www.ling.ohio-state.edu/∼ntyson/tagset/english-tagset.txt.

```
C-PANKOW(document d)
{
    /* recognize all the instances in input document */
    I = recognizeInstances(d);
    foreach i ∈ I
    {
        foreach (p,c) ∈ P
        {
            /* download the n first Google abstracts
            matching the exact query c(i) */
            Abstracts = downloadGoogleAbstracts(c(i),n);
            foreach a in Abstracts
            {
                /* calculate the similarity between the
                document d and the Google abstract a */
                sim = calculateSimilarity(a,d);
                if (sim > t)
                {
                  if (p.matches(a))
                  {
                    c = p.getConcept();
                    Res[c] = Res[c]+sim;
                  }
                }
            }
        }
    }
    annotate(i,maxarg_c Res[c]);
}
```

Figure 2: C-PANKOW's process in pseudocode

(de|la|los|las|del){FW} | [a-z]+{NP|NPS|NN|NNS}

These expressions are intended to be interpreted as standard regular expressions over words and their corresponding part-of-speech tags, which are indicated in curly brackets. Paraphrasing, INSTANCE matches each optional sequence of arbitrary characters (\w+) tagged as a determiner (DT), followed optionally by a sequence of small letters ([a-z]+) tagged as an adjective (JJ), followed by an expression matching the regular expression denoted by PRE, which in turn can be optionally followed by an expression matching the concatenation of MID and POST. Thereby, PRE and POST match a sequence of tokens in which the first character is capitalized and tagged either as a plural proper noun (NPS), a plural common noun (NNS), a common noun (NN), a proper noun (NP), an adjective (JJ) or an interjection UH[3]. MID matches a sequence of determiners 'the', prepositions 'of', the possessive marker ', a hyphen '-', a foreign word FW such as 'de,del,las,los,las' and lower case singular or plural proper and common nouns. For example, the tagged sequence Pas{NP} de{FW} la {FW} Casa{NP} would be recognized as an instance, whereby *Pas* would match the PRE, *de la* the MIDDLE and *Casa* the POST part of the above regular expression. The instances discovered this way in our running example web page are given in Table 1 as a cross 'X' in the S(system) column.

2.2 Downloading Google Abstracts

[3]This is important for processing Asian names which sometimes are tagged as 'UH' by the part-of-speech tagger.

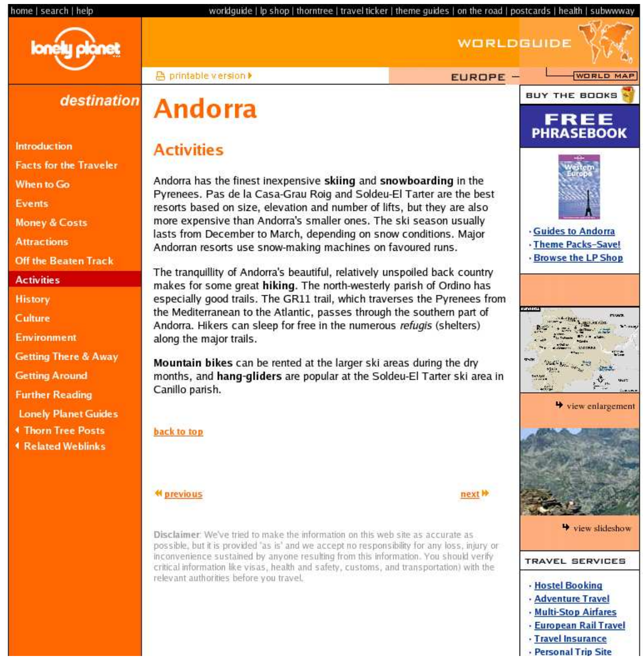

Figure 3: http://www.lonelyplanet.com/destinations/europe/andorra/activities.htm

The patterns in our pattern library are actually tuples (p, c) where p is a regular expression defined over part-of-speech tags as described above, and c a function $c : string \rightarrow string$ called the *clue*. Given an instance $i \in I$ and a clue c, the query $c(i)$ is sent to the Google™ API and we download the abstracts of the first n documents matching this query and then process the abstracts to find instances of pattern p. For example, given the clue $f(x) =$ "$such\ as$ " $\oplus\, x$ and the instance *Seville* we would download n abstracts matching the query f(Seville), i.e. "such as Seville".[4] With the use of such clues, we thus download a number of pages in which a corresponding pattern will probably be matched thus restricting the linguistic analysis to a few promising pages.

2.3 Similarity Assessment

As described in our pseudocode algorithm in Figure 2, we then calculate the similarity between each downloaded abstract and the web page in question. For this purpose, we first remove stopwords from both documents and then adopt the bag-of-words model [30] to create vectors representing the count for each word in the document. Then we use the cosine measure to calculate the similarity between the abstract and the document to be annotated. Thus, we measure the similarity between the abstract and the document as the cosine of the angle between their vectors, i.e.

$$\cos(\sphericalangle(\vec{d}, \vec{a})) = \frac{\vec{d} \cdot \vec{a}}{||\vec{d}|| \cdot ||\vec{a}||}$$

We only consider those pages as relevant for which this similarity is over the threshold t. Further, we weight the contribution of the pattern matched in that page with this value thus 'contextualizing' the pattern-matching process with the result that a pattern matched in a very similar page counts more than a pattern matched in a less similar one. Thus, a certain instance can be annotated with a different concept in different contexts, i.e. web pages. In general,

[4]Here, $\oplus$ denotes the concatenation operator defined on two strings.

the intuition behind this is to yield more accurate annotations and to choose the contextually most appropriate sense or concept for a given instance in case it is ambiguous. Additionally, by this we only linguistically analyze Google abstracts which seem relevant. After a few initial experiments we decided to use '0.05' as threshold value. In section 3 we also present further experiments with different threshold values.

2.4 The Pattern Library

In what follows we present the pattern library P we use and briefly describe the intuition behind each pattern:

2.4.1 Hearst Patterns

These patterns have been applied by Marti Hearst ([23]) to discover sub-/superconcept relations. As we have argued several times in [7] and [8] these patterns can however also be used to discover instance/concept relations. The relations reused from Hearst are the following:

HEARST1:= CONCEPT such{DT} as{IN} (INSTANCE ,?)+ ((and|or){CC} INSTANCE)?

HEARST2:= CONCEPT ,? especially{RB} (INSTANCE ,?)+ ((and|or){CC} INSTANCE)?

HEARST3:= CONCEPT ,? including{RB} (INSTANCE ,?)+ ((and|or){CC} INSTANCE)?

HEARST4:= INSTANCE ,?)+ and{CC} other{JJ} CONCEPT

HEARST5:= INSTANCE ,?)+ or{CC} other{JJ} CONCEPT

where CONCEPT := [a-z]+{NN(S)?}+ and the corresponding clues are:

$\text{clue}_{HEARST1}(x) = such\ as \oplus x$
$\text{clue}_{HEARST2}(x) = especially \oplus x$
$\text{clue}_{HEARST3}(x) = including \oplus x$
$\text{clue}_{HEARST4}(x) = x \oplus and\ other$
$\text{clue}_{HEARST5}(x) = x \oplus or\ other$

An example for an expression matched by the HEARST1 patterns is *hotels such as the Ritz, the Hilton and the Holiday Inn*, one for HEARST3 is *sights, including the Eiffel Tower, the Statue of Liberty or the St. Peter's Chapel*, and one for HEARST4 is *New York, Tokyo, Rio de Janeiro and other big cities*.

2.4.2 Definites

The next pattern involves a definite, i.e. a noun phrase introduced by the definite determiner '*the*'. Frequently, definites actually *refer* to some entity previously mentioned in the text. In this sense, a phrase like '*the hotel*' does not stand for itself, but it points as a so-called anaphora to a unique hotel occurring in the preceding text. Nevertheless, it has also been shown that in common texts more than 50% of all definite expressions are *non-referring* ([29]), i.e. they exhibit sufficient descriptive content to enable the reader to uniquely determine the entity referred to from the global context. For example, the definite description '*the Hilton hotel*' has sufficient descriptive power to uniquely pick-out the corresponding real-world entity for most readers. One may deduce that '*Hilton*' is the name of the real-world entity of type Hotel to which the above expression refers.

DEFINITE: the{DT} (\w+{JJ})? INSTANCE CONCEPT,

whereby the corresponding clue is $\text{clue}_{DEFINITE}(x) = the \oplus x$.

2.4.3 Copula

The probably most explicit way of expressing that a certain entity is an instance of a certain concept is by the verb '*to be*' in a copula[5] construction as for example in '*The Excelsior is a nice hotel in the center of Nancy*'. Here's the general pattern:

INSTANCE \w+{BE(D?)(Z|R)} CONCEPT,

where *is, are, was* and *were* are tagged by the part-of-speech tagger as BEZ, BER, BEDZ and BEDR, respectively. The corresponding clue is $\text{clue}_{COPULA}(x) = x \oplus is$.

2.5 Run Time and Query Size Complexity

The runtime complexity of C-PANKOW is $O(|I| \cdot |P| \cdot n)$, where $|I|$ is the total number of instances to be annotated, $|P|$ the number of patterns we use, and n the maximum number of pages downloaded. As $|P|$ and n are constant and a document of size $|D|$ contains at most $|D|$ instances, the overall complexity of C-PANKOW is thus linear in the size of the document, i.e. $O(|D|)$. As the Google™ API does not allow to retrieve more than 10 documents per query, the number of queries sent to the Google™ API is $|P| \cdot (n\ div\ 10)$ per instance. In our special settings, $|P| = 7$, so that we issue $7\,n\ div$ 10 queries per instance independently of how big the ontology in use is. This is an important reduction of the number of queries compared to PANKOW in which we had to issue $|P'| * |C|$ queries per instance, where $|C|$ indicate the number of concepts the ontology had. For a small set of concepts with $|C| = 59$ as well as $|P'| = 10$ this meant 590 queries per instance to be annotated. As C-PANKOW is independent of the size of the ontology, we can thus consider even larger ontologies than PANKOW, which already provided annotation based on much larger ontologies than most other approaches.

3. EVALUATION

In order to evaluate our system, we have reused the dataset described in [7]. In order to create this dataset, we asked two human subjects to annotate 30 texts with destination descriptions from *http://www.lonelyplanet.com/destinations*. They used a pruned version of the tourism ontology developed within the GETESS project ([32]). The original ontology consisted of 1043 concepts, while the pruned one consisted of 682. The latter ones have been considered also in our evaluation. The subjects were told to annotate instances in the text with the appropriate concept from the ontology. In what follows, we will refer to these subjects as A and B. Subject A actually produced 436 annotations and subject B produced 392. There were 277 instances that were annotated by both subjects. For these 277 instances, they used 59 different concepts and the categorial agreement on these 277 instances as measured by the Kappa statistic was 63.50% (cf. [5]), which allows to conclude that the annotation task is overall well defined but that the agreement between humans is far from perfect. In what follows, we present results for the detection of instances as well as for the actual automatic classification of these.

[5]A copula is an intransitive verb which links a subject to an object, an adjective or a constituent denoting a property of the subject.

3.1 Instance Detection

In order to detect potential instances in a web page to be annotated, we apply the method described in Section 2.1. We apply this method to the dataset described above and compare the instances discovered by our system with the instances annotated by the annotators in terms of Precision, Recall and F-Measure. More formally, let $I_{A,d}$ and $I_{B,d}$ be the set of instances annotated by subjects A and B in the document d, and let $I_{S,d}$ be the set of instances detected by the system in the same document d, then Precision, Recall and F-Measure are defined as follows:

$$P^I_{X,d} = \frac{|I_{X,d} \cap I_{S,d}|}{|I_{S,d}|}$$

$$R^I_{X,d} = \frac{|I_{X,d} \cap I_{S,d}|}{|I_{X,d}|}$$

$$F^I_{X,d} = \frac{2 * P^I_{X,d} * R^I_{X,d}}{P^I_{X,d} + R^I_{X,d}}$$

For these and all the other measures $M \in \{P, R, F, Acc, LA\}$ considered in this section, we average over all the documents and over both annotators, i.e.:

$$M_d = \frac{M_{A,d} + M_{B,d}}{2}$$

$$M = \sum_{d \in D} \frac{M_d}{|D|}$$

In our running example web page, the system for example detected 11 instances, while subject A found 9 and subject B 7 (compare Table 1). The system coincided with subjects A and B in 6 and 5 instances, respectively. This leads to the following results for our running example: $P^I_{A,d} = \frac{6}{11} = 54.55\%$, $P^I_{B,d} = \frac{5}{11} = 45.45\%$, $R^I_{A,d} = \frac{6}{9} = 66.67\%$, $R^I_{B,d} = \frac{5}{7} = 71.43\%$ and thus $F^I_{A,d} = 60\%$, $F^I_{B,d} = 55.55\%$. Thus, $P^I_d = 50\%$, $R^I_d = 69.05\%$ and $F^I_d = 57.78\%$. For the whole dataset, the system achieved a precision of $P^I = 43.75\%$, a recall of $R^I = 57.20\%$ and a F-Measure of $F^I = 48.39\%$. In order to get an upper limit on the task, we also compared the precision, recall and F-measure of subject A against subject B and vice versa and yielded $F^I_{human} = 70.61\%$ as a human baseline on the task. While we are still quite far away from the human performance on the task, the results are as desired in the sense that we have a higher recall at the cost of a lower precision. This is useful as some of the spurious instances will be filtered out by the instance classification step due to the fact that if no patterns are found, the instance will not be assigned to any concept. Thus, the precision can be increased by the instance classification step, while the recall needs to be reasonably high as it can not be increased later.

3.2 Instance Classification

The annotations by the subjects A,B as well as by the system for a document d are modeled by the functions $f_{A,d}$, $f_{B,d}$ and $f_{S,d}$, respectively. The actual classification of the detected instances is also evaluated in terms of Precision, Recall and F-measure with respect to both reference sets A and B. For this purpose we introduce a set C of instance/concept pairs as follows:

$$C_{X,d} = \{(i,c) | i \in dom(f_{X,d}) \ and \ f_{X,d}(i) = c)\}$$

Precision, Recall and F-Measure are defined as follows:

$$P^C_{X,d} = \frac{|C_{X,d} \cap C_{S,d}|}{|C_{S,d}|}$$

Instance	A	B	S
Andorra	X	X	X
Andorra Activities Andorra			X
Atlantic	X	X	X
Canillo	X	X	X
GR11	X		X
Lonely Planet World			X
Major Andorran			X
Mediterranean	X	X	
Ordino	X	X	X
Pas de la Casa	X		
Pyrenees	X	X	X
Roig			X
Soldeu	X		
Soldeu-El Tarter			X
Traveler		X	

Table 1: Results of the instance detection algorithm

$$R^C_{X,d} = Acc_{X,d} = \frac{|C_{X,d} \cap C_{S,d}|}{|C_{X,d}|}$$

$$F^C_{X,d} = \frac{2 * P^C_{X,d} * R^C_{X,d}}{P^C_{X,d} + R^C_{X,d}}$$

It is important to mention that our recall corresponds to the accuracy used in other approaches (compare [2] or [22]), such that we can compare with these using this value. As above these measures are averaged over both annotators and over the 30 documents. In our running example for instance, the system produced 6 annotations, while subject A and B produced the above mentioned 9 and 7 annotations, respectively. The system agrees with A and B in 2 annotations, respectively (compare Table 2). This leads to the following precision, recall and F-Measure values: $P^C_{A,d} = \frac{2}{6} = 33.33\%$, $R^C_{A,d} = \frac{2}{9} = 22.22\%$, $F^C_{A,d} = 26.66\%$ and $P^C_{B,d} = \frac{2}{6} = 33.33\%$, $R^C_{B,d} = \frac{2}{7} = 28.57\%$, $F^C_{B,d} = 30.77\%$. Thus, $P^C_d = 33.33\%$, $R^C_d = 25.40\%$, $F^C_d = 28.72\%$.

Furthermore, to assess how good the actual classification is, we also consider the Accuracy of the system which abstracts from the actual instance recognition task. For this, we present two further measures of accuracy. The first one (Acc') considers the instances annotated by the system as well as by the human thus not penalizing the system for not giving answers for instances it didn't discover. The second one, Acc'', considers the 277 common instances annotated by both subjects in order to compare our results with our earlier system presented in [7]. Here follow the formal definitions:

$$Acc'_{X,d} = \frac{|C_{X,d} \cap C_{S,d}|}{|dom(f_{X,d}) \cap dom(f_{S,d})|}$$

$$Acc''_{X,d} = \frac{|\{(i,c) | i \in dom(f_{A,d}) \cap dom(f_{B,d}) \wedge f_{X,d} = c\} \cap C_{S,d}|}{|dom(f_{A,d}) \cap dom(f_{B,d})|}$$

Thus we get for our running example: $Acc'_{A,d} = Acc'_{B,d} = Acc'_d = \frac{2}{5} = 40\%$ and $Acc''_{A,d} = Acc''_{B,d} = Acc''_d = \frac{2}{6} = 33.33\%$.

Finally, as instances 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 et al. [22]. However, we consider a slightly different formulation of the Learning Accuracy in line with the measures defined in [26]. Both measures are in fact equivalent, the only 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. First of all we need the notion of the least common superconcept c of two concepts a and b which is defined in line with [26]:

$$lcs(a, b) := c \text{ such that } \delta(a, c) + \delta(b, c) + \delta(top, c) \text{ is minimal}$$

Now the taxonomic similarity T_{sim} between two concepts is defined as:

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

where $c = lcs(a, b)$.
The Learning Accuracy is now defined as follows:

$$LA_{X,d} = \sum_{i \in dom(f_{X,d}) \cap dom(f_{S,d})} \frac{T_{sim}(f_{S,d}(i), f_{X,d}(i))}{|dom(f_{S,d}) \cap dom(f_{X,d})|}$$

Here we also average over both annotators and over the documents in the collection.

3.2.1 Threshold

In a first series of experiments, we examined the effect of varying the threshold, but without taking into account the similarity for weighting the patterns. As the results in Table 3 show, the best results were in fact achieved when using a threshold of 0.05. When increasing the threshold, the precision of the annotations certainly increases, but the recall is drastically reduced. In general, all values except for the precision decrease when increasing the threshold such that we conclude that the best threshold lies somewhere between 0 and 0.1. Thus a threshold of 0.05 seems reasonable.

3.2.2 Similarity

Table 4 shows the results for the baseline experiment with no threshold, i.e. considering all the pages returned by Google up to a maximum of 100 (labeled with 'no threshold' in the table). It also shows the results of the experiment using a threshold of 0.05 as well as one in which the patterns were weighted according to the corresponding similarity. The results already show that taking into account the similarity and considering only those pages with a similarity over the threshold t indeed yields better results. Further, the version weighting the patterns yields a higher precision at the cost of a slightly lower recall, but increases the Learning Accuracy by more than 3 points. Thus, we conclude that using the version of our system weighting the patterns according to the similarity of the involved pages indeed yields better results.

3.2.3 Number of Pages

Finally, we also varied the maximum number of abstracts downloaded. The results are given in Table 5. Interestingly, using a lower or higher number of maximum pages also decreased the results independently if the contribution of each pattern was weighted according to the similarity between the page to be annotated and the corresponding Google abstract or not. In general, we conclude that using maximally the first 100 hits returned by Google for the clue patterns is enough.

3.2.4 A posteriori evaluation

In order to apply our approach to a larger set of web pages, we selected a set of 307 news stories from *http://news.kmi.open.ac.uk/rostra/*. These 307 news stories were automatically annotated using our C-PANKOW web service using the first 100 pages returned by Google, a threshold of 0.05 and taking into account the similarity of the abstract in which the pattern was matched. In this way 1270 annotations were produced, i.e. 4.1 annotations on average per document. One of our annotators manually analyzed the annotations *a posteriori* and evaluated each annotation by assigning a value from 0 (incorrect) to 3 (totally) correct. We are thus only able to give results for the precision of our system on this dataset. We measure three types of Precision: P_3, P_2 and P_1 considering an answer as correct if it was assigned at least 3, 2 and 1 points, respectively. On average, the annotations produced by the system received 1.81 points by the annotator. The precisions were: $P_3 = 54.88\%$, $P_2 = 57.95\%$ and $P_1 = 68.66\%$. In order to compare these results with the results on the Lonely Planet dataset, we also performed this *a posteriori* evaluation on that dataset yielding 2.1 points per annotation on average and the following precisions: $P_3 = 58.14\%$, $P_2 = 71.1\%$ and $P_1 = 76.08\%$. These results corroborate the fact that the annotations produced by C-PANKOW are indeed very accurate.

3.2.5 WordNet as Ontology

As a final experiment, instead of annotating with respect to a domain-specific ontology, we used a general purpose ontology, i.e. WordNet [18]. An additional complication here is that words can have different meanings or *senses* – as they are called in WordNet terminology – and thus the correct meaning needs to be chosen on the basis of contextual evidence. For this purpose we implemented a simple word sense disambiguation algorithm in line with the one presented in [25]. In fact, in our approach we choose that sense whose gloss maximizes the overlap – in the number of words – with the web page in question. In order to evaluate the annotations with respect to WordNet, we conducted again a posteriori evaluation on the 307 news stories as mentioned above. The human subject was asked to evaluate the appropriateness of the annotation on the basis of the synset's gloss, i.e. a natural language description of its intention. For this task, the results were actually poorer with precisions of $P_3 = 27.91\%$, $P_2 = 33.47\%$ and $P_1 = 43.43\%$. Further we identified two main reasons why the results are lower with respect to using domain-specific ontology. First, in some cases the term with maximal evidence, i.e. with the largest number of Google hits, is not specific to the domain in question, but as WordNet is so large, a corresponding concept or synset is almost always found and the named entity in question thus annotated with it. Second, in other cases the term with maximal evidence is contextually appropriate, but our simple word sense disambiguation approach fails in selecting the correct sense. We thus conclude that in order to accurately annotate with respect to WordNet we would need a mechanism to consider only relevant terms for the domain in question as well as to improve our word sense disambiguation algorithm.

3.2.6 Discussion & Comparison

We have presented results showing that the use of the similarity as an indicator of the relevance of a certain Google abstract for the page in question indeed improves the quality of the annotations produced by the system. Further, we have also examined two parameters: the similarity threshold and the maximum number of result pages considered. The results show that a similarity threshold of 0.05 seems reasonable. Furthermore, an interesting result is that increasing the number of pages considered does not improve the quality of the annotations. This is a very important result from a practical point of view as it shows that we can get good results while maintaining efficiency at the same time.

In order to assess the performance of our system we compare it

Instance	A	B	S
Andorra	country	country	country
Atlantic	sea	sea	fish
Canillo	town	area	hotel
GR11	walking_trail		
Mediterranean	sea	sea	
Ordino	region	area	valley
Pas de la Casa	town		
Pyrenees	mountain	mountain	mountain
Roig			family
Soldeu	town		
Traveler		person	

Table 2: Annotations by subjects A, B and the system (S)

Threshold	F^C	P^C	R^C=Acc	Acc'	Acc"	LA
no threshold	19.64%	23.27%	17.83%	41.34%	27.91%	68.03%
0.05	**22.31%**	29.15%	**18.96%**	46.19%	**29.32%**	71.31%
0.1	15.19%	36.18%	10.58%	50.44%	19.21%	76.82%
0.3	1.41%	60.83%	0.79%	79.17%	7.22%	89.92%
0.5	1.41%	60.83%	0.79%	79.17%	7.22%	89.92%
0.7	1.41%	60.83%	0.79%	79.17%	7.22%	89.92%
0.9	1.41%	60.83%	0.79%	79.17%	7.22%	89.92%

Table 3: Results of varying the threshold (no weighting,n=100)

Threshold	F^C	P^C	R^C=Acc	Acc'	Acc"	LA
no threshold	19.64%	23.27%	17.83%	41.34%	27.91%	68.03%
t=0.05	**22.31%**	29.15%	**18.96%**	46.19%	29.32%	71.31%
t=0.05 + sim	22.27%	**29.29%**	18.92%	**46.79%**	**29.35%**	**74.37%**

Table 4: Impact of using the similarity measure (n=100)

	No Weighting					
No. pages	F^C	P^C	R^C=Acc	Acc'	Acc"	LA
10	17.38%	29.47%	13.73%	42.54%	23.57%	69.92%
50	20.72%	**31.03%**	17.40%	43.49%	27.43%	**71.65%**
100	**22.31%**	29.15%	**18.96%**	**46.19%**	**29.32%**	71.31%
200	18.27%	19.89%	17.70%	43.86%	28.13%	70.71%
300	17.78%	19.03%	17.47%	41.15%	27.58%	70.09%
	Weighting					
No. pages	F^C	P^C	R^C=Acc	Acc'	Acc"	LA
10	16.26%	30.38%	12.27%	43.58%	21.59%	72.28%
50	20.76%	**31.53%**	17.33%	45.45%	27.43%	73.47%
100	**22.27%**	29.29%	18.92%	**46.79%**	29.35%	**74.37%**
200	19.62%	21.39%	**18.97%**	45.97%	**29.50%**	71.49%
300	18.47%	19.88%	18.09%	43.54%	28.30%	72.17%

Table 5: Results of varying the number of pages (t=0.05)

to state-of-the-art systems performing the same task, i.e. assigning instances appearing in texts to their corresponding concept in a given ontology. The approaches we directly compare with are the ones in [2] and [22] as they consider a similar scenario. However, we also situate our approach in the context of other named entity recognition and classification approaches. In particular, we discuss other systems with a special emphasis on the number of concepts considered, which renders a classification easier or more difficult. Table 6 shows the results of different systems ordered according to number of concepts considered for the assignment. The systems are described in more detail in Section 5.

4. IMPLEMENTATION

C-PANKOW has been implemented in Java and is accessible as a Web Service using Axis[6], an open implementation of the Simple Object Access Protocol (SOAP)[7], as well as a Servlet on top of Tomcat[8] with a web frontend. Thus, C-PANKOW can be either accessed in a programmatic way or alternatively through the web frontend at http://km.aifb.uni-karlsruhe.de/pankow/annotation/. In C-PANKOW, KAON [14], an open source ontology management infrastructure developed at our institute is used to represent ontologies and perform ontology validation. The Google index is searched using the Google API, which is available from Google. Our implementation uses the free-of-charge Google API, which only yields ten results per search. For efficiency reasons, this queries to Google are multi-threaded with up to ten simultaneous searches. Furthremore, as the default timeout of the Google API is relatively high, time limits are imposed on each thread after which the thread will be restarted. Because annotating one page can take up to 20 minutes, the server has a dedicated worker thread that does the actual annotation using the methods put forth in this paper. Both the Servlet and the Axis interface enter the user's request into a queue which is then emptied by the worker thread. The results are then dispatched to the user via email, although other methods of notification are possible. Currently, C-PANKOW is integrated with the semantic web browser MagPie [13] with the purpose of providing additional named entities to be highlighted.

5. RELATED WORK

As many others, our work is based on the seminal work of Hearst on applying hand-crafted patterns denoting a certain relation to find instances of these relations in a text corpus. Hearst applied such relations to discover is-a relations in text. Hearst's idea has been reapplied by different researchers with either slight variations in the patterns used [24], in very specific domains [1], to acquire knowledge for anaphora resolution [28], or to discover other kinds of semantic relations such as part-of relations [6] or causation relations [21]. Instead of matching these patterns in a large text collection, some researchers have recently turned to the Web to match these patterns such as in [7], [15], [27]. Especially interesting in our context is the work in [15] which aim is to acquire instances for a given concept. In particular, Etzioni et al. present results on the task of acquiring instances of cities, countries, US states, films and actors. They make use of a Bayesian classifier in order to decide weather an instance belongs to a certain concept or not. Recently, they have also considered learning new patterns by a rule learning process [16]. Though their work is quite similar to ours, the aims of both approaches are quite orthogonal, i.e. while we are concerned with annotating the instances in a given document with the corresponding concept, Etzioni et al. aim at learning the complete extension of a certain concept in order to build a search engine 'knowing it all'. Thus a crucial aspect of the approach presented in this paper is to account for the context in which a certain instance appears in order to annotate it with the contextually most appropriate concept. This is an aspect totally neglected by the approach of Etzioni et al., such that it is doubtful if their approach could be used for an annotation task as we consider here. Furthermore, as their aim is to learn the extension of certain concepts such as actors, cities etc. and thus their task quite different to ours, we do not give any quantitative comparison.

Another interesting system is SemTag [12] which also automatically annotates web pages. Though its results are certainly impressive, it is important to note that SemTag actually only performs the task of disambiguating entities appearing in a web page as it relies on the TAP lexicon to list all the possible concepts, senses or meanings of a given entity. Our approach does not rely on such a handcrafted lexicon and in fact automatically discovers all the possible concepts for a given entity from the Web and then suggests the contextually most appropriate concept for it thus performing two tasks in one: induction of possible concepts and context-based disambiguation. Thus, as the tasks are not comparable, we also do not provide any quantitative comparison of results.

Brin [3] presents a bootstrapping approach in which the system starts with a few patterns, and then tries to induce new patterns using the results of the application of the seed patterns as training dataset. This is also the general idea underlying the Armadillo system [10], which exploits redundancy in the World Wide Web to induce such extraction rules. The work of [11] is also concerned with inducing certain patterns to extract information, but does this by learning 'soft patterns' and relying on a similarity criterion when matching these. This is quite different from the 'hard patterns' that are considered in the approaches such as the one presented in this paper as well as [9] or [15].

Concerning the task of learning the correct class or ontological concept for an unknown entity, there is some related work, especially in the computational linguistics community. In the context of the Message Understanding Conferences (MUC), systems typically achieved accuracies of well above 90% in the task of tagging named entities with respect to three classes: *organization*, *person* and *location*. However, this task is quite moderate compared to the task of using 682 concepts as considered in our approach. Other systems have also considered more categories such as Hovy et al. [19] which considered 7 or Evans [17] which considered from 2-8 categories varying form document to document as well as Hahn et al. [22] and Alfonseca et al. [2] who consider 325 and 1200 concepts, respectively.

Hahn and Schnattinger [22] create a *hypothesis space* when encountering an unknown word in a text for each concept that the word could belong to. These initial hypothesis spaces are then iteratively refined on the basis of evidence extracted from the linguistic context the unknown word appears in. In their approach, evidence is formalized in the form of quality labels attached to each hypothesis space. At the end the hypothesis space with maximal evidence with regard to the qualification calculus used is chosen as the correct ontological concept for the word in question. The results of the different version of Hahn et al.'s system (compare [22]) in terms of accuracy can be found in Table 6. Their approach is very related to ours and in fact they use similar patterns to identify instances from the text. However, the approaches cannot be directly compared. The reason is that they evaluate their approach under clean room conditions as they assume accurately identified syntactic and

[6]http://ws.apache.org/axis
[7]http://www.w3.org/TR/soap/
[8]http://jakarta.apache.org/tomcat/

System	No. Concepts	Preprocessing	Accuracy/Recall	Learning Accuracy
MUC	3	various	>90%	n.a
Evans	2-8	typology derivation (clustering)	41.41%	n.a.
Fleischman et al.	8	N-gram frequency extraction	70.4%	n.a.
PANKOW	59	none	24.9%	58.91%
Hahn et al. (Baseline)	325	perfect syntactic and semantic analysis required	21%	67%
Hahn et al. (TH)	325	perfect syntactic and semantic analysis	26%	73%
Hahn et al. (CB)	325	perfect syntactic and semantic analysis	31%	76%
C-PANKOW	682	POS tagging & pattern-matching	29.35%	74.37%
Alfonseca et al. (Object)	1200	syntactic analysis	17.39%	44%

Table 6: Comparison of results

semantic relationships and an elaborate ontology structure, while our evaluation is based on very noisy real-world input — rendering our task harder than theirs. Furthermore, while our evaluation was conducted with respect to a reference standard, it is not clear how they evaluated their system.

Alfonseca and Manandhar [2] have also addressed the problem of assigning the correct ontological class to unknown words. Their system is based on the distributional hypothesis, i.e. that words are similar to the extent to which they share linguistic contexts. In this line, they adopt a vector-space model and exploit certain syntactic dependencies as features of the vector representing a certain word. The unknown word is then assigned to the category corresponding to the most similar vector. The best result measured against a reference standard (strict evaluation mode as they call it) was achieved using only verb/object dependencies as features (compare Table 6). Fleischmann and Hovy [19] address the classification of named entities into fine-grained categories. In particular, they categorize named entities denoting persons into the following 8 categories: *athlete*, *politician/government*, *clergy*, *businessperson*, *entertainer/ artist*, *lawyer*, *doctor/scientist*, *police*. Given this categorization task, they present an experiment in which they examine 5 different Machine Learning algorithms: C4.5, a feed-forward neural network, k-nearest Neighbors, a Support Vector Machine and a Naive Bayes classifier. As features for the classifiers they make use of the frequencies of certain N-grams preceding and following the instance in question as well as topic signature features which are complemented with synonymy and hyperonymy information from WordNet. They report a best result of an accuracy of 70.4% when using the C4.5 decision tree classifer. Fleischman and Hovy's results are certainly very high in comparison to ours – and also to the ones of Hahn et al. [22] and Alfonseca et al. [2] – but on the other hand though they address a harder task than the MUC Named Entity Task, they are still quite far away from the number of categories we consider here.

Evans [17] derives similar statistical fingerprints as considered in our approach by querying Google™ and then clusters named entities on the basis of these fingerprints as features in order to derive a class topology from the document in question. He uses a bottom-up hierarchical clustering algorithm for this purpose. His approach differs from the others discussed here in that it is totally unsupervised without even the set of categories being given. Thus, the entities are classified with respect to different sets of categories depending on the document considered. Overall, he reports 41.41% of correctly classified entities, considering from 2 to 8 classes.

6. CONCLUSION

We have presented an enhancement of our original PANKOW approach called C-PANKOW. With C-PANKOW we have overcome several shortcomings of the earlier system. In fact, C-PANKOW has outperformed PANKOW as well as some of its closest competitors. First, by downloading a certain number of pages, linguistically analyzing and normalizing them, we overcome problems of our earlier pattern generation method and increase the recall of the pattern matching process by being able to consider structures with a more complex linguistic structure such as noun phrases containing determiners, adjectives or other noun modifiers. At the same time we reduce the number of queries sent to the Google API which was originally proportional to the size of the ontology. Instead of generating patterns for all the concepts in the ontology, we match the patterns in the downloaded Google abstracts and map the results to the ontology in question such that the approach is independent of the ontology's size, thus being scalable to arbitrarily large ontologies. In fact, the complexity of our approach is now linear in the number of instances and hence in the size of the document. The number of queries sent to the Google API is now constant for each instance to be annotated. Finally and most important, by considering the similarity between the page to be annotated and the Google abstract in which the pattern was matched we have contextualized our approach to provide provably more accurate annotations in the context of the document to be annotated and to choose the contextually most appropriate concept for a possibly ambiguous instance. Our results have shown that considering such a similarity indeed improves the quality of the results of our system, thus being one step closer towards the 'Self-Annotating Web'.

Our work has been so far restricted to learning annotations for instances. Further work will extend this to also learning to annotate conceptual relations between discovered instances. A very interesting direction for further research would be to learn new patterns indicating a certain relation by a certain rule induction process such as in [3, 9, 11, 16]. The named entity recognition in Web sites could also be improved.

Acknowledgments. We would like to thank Google for their support with their Web service API. Thanks also to all our colleagues for their support, in particular to Siegfried Handschuh and Kai Kühn for integrating PANKOW with our in-house annotation tool OntoMat as well as to Martin Dzbor for helping to integrate C-PANKOW with MagPie. Further, we would also like to acknowledge Ursula Sciesinski and Laura Goebes for annotating the Lonely Planet Web pages and the latter for performing the a posteriori evaluation with respect to the KMI stories and Lonely Planet datasets. Thank also to Victoria Uren from KMI for providing this dataset. Research for this paper has been funded by the IST-Dot.Kom project (grant IST-2001-34038). The Dot.Kom project (http://www.dot-kom.org) is sponsored by the EC as part of the framework V.

7. REFERENCES

[1] K. Ahmad, M. Tariq, B. Vrusias, and C. Handy. Corpus-based thesaurus construction for image retrieval in specialist domains. In *Proceedings of the 25th European Conference on Advances in Information Retrieval (ECIR)*, 2003.

[2] E. Alfonseca and S. Manandhar. Extending a lexical ontology by a combination of distributional semantics signatures. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, pages 1–7, 2002.

[3] S. Brin. Extracting patterns and relations from the World Wide Web. In *Proceedings of the WebDB Workshop at EDBT '98*, 1998.

[4] M.E. Califf and R.J. Mooney. Bottom-up relational learning of pattern matching rules for information extraction. *Machine Learning Research*, 4(2):177–210, 2004.

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

[6] E. Charniak and M. Berland. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 57–64, 1999.

[7] P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Proceedings of the 13th World Wide Web Conference*, 2004.

[8] P. Cimiano and S. Staab. Learning by googling. *SIGKDD Explorations*, 6(2), December 2004.

[9] F. Ciravegna. Adaptive information extraction from text by rule induction and generalization. In *Proceedings of tht 17th International Joint Conference on Artificial Intelligence (IJCAI)*, 2001.

[10] F. Ciravegna, A. Dingli, D. Guthrie, and Y. Wilks. Integrating Information to Bootstrap Information Extraction from Web Sites. In *Proceedings of the IJCAI Workshop on Information Integration on the Web*, pages 9–14, 2003.

[11] H. Cui, M.-Y. Kan, and T.-S. Chua. Unsupervised learning of soft patterns for generating definitions from online news. In *Proceedings of the 13th World Wide Web Conference*, pages 90–99, 2004.

[12] 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 World Wide Web Conference*, pages 178–186. ACM Press, 2003.

[13] M. Dzbor, E. Mota, and J. Domingue. Opening up magpie via semantic services. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, 2004.

[14] E. Bozsak et al. KAON - Towards a large scale Semantic Web. In *Proceedings of the Third International Conference on E-Commerce and Web Technologies (EC-Web)*. Springer Lecture Notes in Computer Science, 2002.

[15] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. Web-scale information extraction in KnowItAll (preliminary results). In *Proceedings of the 13th World Wide Web Conference*, pages 100–109, 2004.

[16] O. Etzioni, M. Cafarella, D. Downey, A-M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. Methods for domain-independent information extraction from the web: An experimental comparison. In *Proceedings of the AAAI Conference*, 2004.

[17] R. Evans. A framework for named entity recognition in the open domain. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP)*, pages 137–144, 2003.

[18] C. Fellbaum. *WordNet, an electronic lexical database*. MIT Press, 1998.

[19] M. Fleischman and E. Hovy. Fine grained classification of named entities. In *Proceedings of the Conference on Computational Linguistics, Taipei, Taiwan, August 2002*, 2002.

[20] F. Freitag and N. Kushmerick. Boosted Wrapper Induction. In *Proceedings of the AAAI Conference*, pages 577–583, 2000.

[21] R. Girju and M. Moldovan. Text mining for causal relations. In *Proceedings of the FLAIRS Conference*, pages 360–364, 2002.

[22] U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *AAAI'98/IAAI'98 Proceedings of the 15th National Conference on Artificial Intelligence and the 10th Conference on Innovative Applications of Artificial Intelligence*, 1998.

[23] M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, 1992.

[24] L.M. Iwanska, N. Mata, and K. Kruger. Fully automatic acquisition of taxonomic knowledge from large corpora of texts. In L.M. Iwanksa and S.C. Shapiro, editors, *Natural Language Processing and Knowledge Processing*, pages 335–345. MIT/AAAI Press, 2000.

[25] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26, 1986.

[26] A. Maedche, V. Pekar, and S. Staab. Ontology learning part one - on discovering taxonomic relations from the web. In *Web Intelligence*. Springer, 2002.

[27] K. Markert, N. Modjeska, and M. Nissim. Using the web for nominal anaphora resolution. In *EACL Workshop on the Computational Treatment of Anaphora*, 2003.

[28] M. Poesio, T. Ishikawa, S. Schulte im Walde, and R. Viera. Acquiring lexical knowledge for anaphora resolution. In *Proceedings of the 3rd Conference on Language Resources and Evaluation (LREC)*, 2002.

[29] M. Poesio and R. Vieira. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216, 1998.

[30] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[31] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272, 1999.

[32] S. Staab, C. Braun, I. Bruder, A. Düsterhöft, A. Heuer, M. Klettke, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, and B. Wrenger. Getess - searching the web exploiting german texts. In *Proceedings of the 3rd Workshop on Cooperative Information Agents*. Springer Verlag, 1999.