Towards the Self-Annotating Web

Philipp Cimiano[1], Siegfried Handschuh[1], Steffen Staab[1,2]
[1]Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
{pci, sha, sst}@aifb.uni-karlsruhe.de
http://www.aifb.uni-karlsruhe.de/WBS
[2]Ontoprise GmbH, 76131 Karlsruhe, Germany
http://www.ontoprise.com/

ABSTRACT

The success of the Semantic Web depends on the availability of ontologies as well as on the proliferation of web pages annotated with metadata conforming to these ontologies. Thus, a crucial question is where to acquire these metadata. In this paper we propose PANKOW (Pattern-based Annotation through Knowledge on the Web), a method which employs an unsupervised, pattern-based approach to categorize instances with regard to an ontology. The approach is evaluated against the manual annotations of two human subjects. The approach is implemented in OntoMat, an annotation tool for the Semantic Web and shows very promising results.

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

Measurement, Documentation, Design, Experimentation, Human Factors, Languages

Keywords

Semantic Annotation, Metadata, Information Extraction, Semantic Web

1. INTRODUCTION

The Semantic Web builds on contents that are described semantically via ontologies and metadata conforming to these ontologies. While one sees a plenitude of ontologies and ontology-like structures being defined in research projects like DAML[1], in standardization efforts like the ones by OASIS[2] and in industrial endeavors like UDDI[3], corresponding metadata are mostly missing.

The reason is that in spite of methods and tools like large-scale information extraction [9], learning of information extraction rules [7], and the application of both in current annotation toolsets [17, 28], the obstacles for producing such markup remain high:

- Manual definition of an information extraction system is a laborious task requiring a lot of time and expert know-how ([3]); and
- Learning of extraction rules requires a lot of, frequently too many, examples for learning the rules.

Here, one encounters a vicious circle where there is no Semantic Web because of a lack of metadata, and there are no metadata, because there is no Semantic Web that one could learn from, for example, by training an IE system such as Amilcare ([7]).

As a way out of this vicious cycle we propose a new paradigm: the *Self-annotating Web*. The principal idea of the self-annotating Web is that it uses globally available Web data and structures to semantically annotate — or at least facilitate annotation of — local resources. Initial blueprints for this paradigm are found in such works as the following:

- Some researchers use explicit, linguistically motivated natural-language descriptions to propose semantic relationships ([6, 13, 19, 22]).
- Others use the Web to cope with data sparseness problems in tasks that require statistics about possible semantic relationships ([1, 14, 21, 23]).
- In [10, 12], the Web structure itself is used to determine a focus for harvesting data. Thus, specialized semantic relationships, such as recommendations coming from a particular Web community, can be derived.

Going a step towards the Semantic Web, we propose an original method called *PANKOW* (Pattern-based Annotation through Knowledge On the Web), which employs an unsupervised, pattern-based approach to categorize instances with regard to a given ontology.

The approach is novel, combining the idea of using linguistic patterns to identify certain ontological relations as well as the idea of using the Web as a big corpus to overcome data sparseness problems. It is unsupervised as it does not rely on any training data annotated by hand and it is pattern-based in the sense that it makes use of linguistically motivated regular expressions to identify instance-concept relations in text. The driving principle behind PANKOW is one of *disambiguation by maximal evidence* in the sense that for a given instance it proposes the concept with the maximal evidence derived from Web statistics. The approach thus bootstraps semantic annotations as it queries the Web for relevant explicit natural-language descriptions of appropriate ontological relations.

PANKOW has been conceived for our annotation framework CREAM [16] and has been implemented in OntoMat[4] using queries to the Web service API of Google™. The automatic annotation produced by PANKOW has been evaluated against semantic annotations produced by two independent human subjects.

[1] `www.daml.org/ontologies/`

[2] Consider, e.g., nomenclatures like Universal Business Language or HumanMarkup `http://www.oasis-open.org`.

[3] `www.uddi.org`

[4] `annotation.semanticweb.org/tools/ontomat`

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-844-X/04/0005.

The structure of this paper is as follows: Section 2 describes the principal procedure of PANKOW. Section 3 describes the core algorithmic approach to categorizing instances from text. In Section 4, we present the empirical results of our evaluation. Then, we briefly discuss the integration into CREAM/OntoMat (Section 5). Before concluding the paper, we discuss related work in Section 6.

2. THE PROCESS OF PANKOW

This section gives a general overview of the process of PANKOW whereas Section 3 explains the concrete methods and Section 5 the implementation details. The process consists of four steps (depicted in Figure 1):

Input: A web page.

In our implementation, we assume that Web pages are handled individually in the CREAM/OntoMat framework ([18]), though actually batch processing of a whole Web site would be possible.

Step 1: The system scans the Web page for phrases in the HTML text that might be categorized as instances of the ontology. Candidate phrases are proper nouns, such as '*Nelson Mandela*', '*South Africa*', or '*Victoria Falls*'). We use a part-of-speech tagger (cf. Section 3 and Section 5) to find such candidate proper nouns.

Thus, we end up with a

Result 1: set of candidate proper nouns

Step 2: The system iterates through the candidate proper nouns. It uses the approach described in Section 3.1, introducing all candidate proper nouns and all candidate ontology concepts into linguistic patterns to derive hypothesis phrases. For instance, the candidate proper noun '*South Africa*' and the concepts Country and Hotel are composed into a pattern resulting in hypothesis phrases like '*South Africa is a country*' and '*South Africa is a hotel*'.

Result 2: Set of hypothesis phrases.

Step 3: Then, Google™ is queried for the hypothesis phrases through its Web service API (Section 3.2). The API delivers as its results

Result 3: the number of hits for each hypothesis phrase.

Step 4: The system sums up the query results to a total for each instance-concept pair. Then the system categorizes the candidate proper nouns into their highest ranked concepts (cf. Section 3.3). Hence, it annotates a piece of text as describing an instance of that concept. Thus we have

Result 4: an ontologically annotated web page.

In principle, the query results of step 3 could be investigated further. For instance, it could make sense to constrain the number of hits for hypothesis phrases to the ones that occur in Web pages with topics closely related to the topic of the current Web page, as, e.g. measured in terms of cosine distance of the documents. However, without direct access to the Google™ databases we have considered this step too inefficient for use in automatic annotation and hence ignore it in the following.

3. PATTERN-BASED CATEGORIZATION OF CANDIDATE PROPER NOUNS

There is some history of applying linguistic patterns to identify ontological relationships between entities referred to in a text. For instance, Hearst [19] as well as Charniak and Berland [6] make use of such a pattern-based approach to discover taxonomic and part-of relations from text, respectively. Hahn and Schnattinger [15] also make use of such patterns and incrementally established background knowledge to predict the correct ontological class for unknown named entities appearing in a text. The core idea of any such pattern-based approach is that one may justify an ontological relationship with reasonable accuracy when one recognizes some specific idiomatic/syntactic/semantic relationships. Germane to the pattern-based approach is that the specifically addressed idiomatic/syntactic/semantic relationships may be very easily spotted because they may be typically specified through simple and efficiently processable regular expressions.

In the following, we first present the set of patterns, in a second step we describe the procedure to actually search for them and finally we explain how we use the information conveyed by them for the actual classification of instances.

3.1 Patterns for Generating Hypothesis Phrases

In the following we describe the patterns we exploit and give a corresponding example from the data set that we used for empirical evaluation (cf. Section 4).

3.1.1 Hearst Patterns

The first four patterns have been used by Hearst to identify *isa*-relationships between the concepts referred by two terms in the text. However, they can also be used to categorize a candidate proper noun into an ontology.

Since the entities denoted by candidate proper nouns are typically modeled as instances of an ontology, we also describe the problem more conveniently as the instantiation of a concept from a given ontology. Correspondingly, we formulate our patterns using the variable '<INSTANCE>' to refer to a candidate noun phrase, as the name of an ontology instance, and '<CONCEPT>' to refer to the name of a concept from the given ontology.

The patterns reused from Hearst are:

H1: <CONCEPT>s such as <INSTANCE>

H2: such <CONCEPT>s as <INSTANCE>

H3: <CONCEPT>s, (especially|including)<INSTANCE>

H4: <INSTANCE> (and|or) other <CONCEPT>s

The above patterns would match the following expressions (in this order): *hotels such as Ritz*; *such hotels as Hilton*; *presidents, especially George Washington*; and *the Eiffel Tower and other sights in Paris*.

3.1.2 Definites

The next patterns are about definites, i.e. noun phrases 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

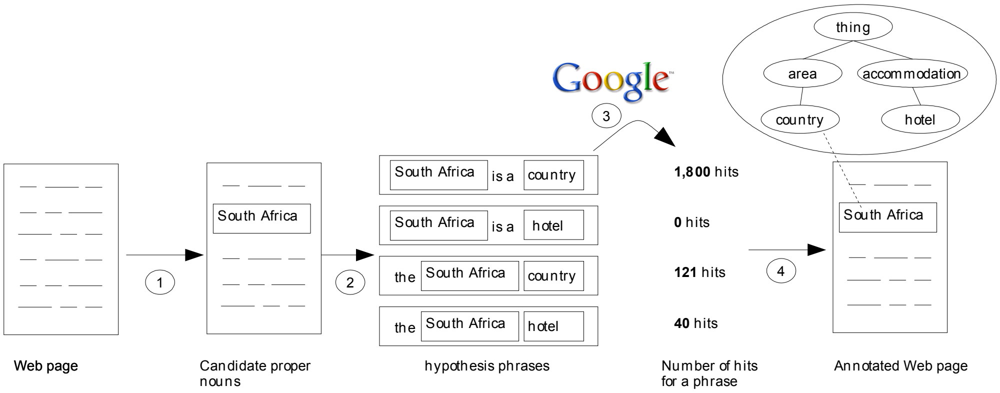


Figure 1: The Process of PANKOW

more than 50% of all definite expressions are *non-referring*, i.e. they exhibit sufficient descriptive content to enable the reader to uniquely determine the entity referred to from the global context ([24]). 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.

Consequently, we apply the following two patterns to categorize candidate proper nouns by definite expressions:

DEFINITE1: the <INSTANCE> <CONCEPT>

DEFINITE2: the <CONCEPT> <INSTANCE>

The first and the second pattern would, e.g., match the expressions '*the Hilton hotel*' and '*the hotel Hilton*', respectively.

3.1.3 Apposition and Copula

The following pattern makes use of the fact that certain entities appearing in a text are further described in terms of an apposition as in '*Excelsior, a hotel in the center of Nancy*'. The pattern capturing this intuition looks as follows:

APPOSITION: <INSTANCE>, a <CONCEPT>

Probably, the most explicit way of expressing that a certain entity is an instance of a certain concept is by the verb '*to be*', as for example in '*The Excelsior is a hotel in the center of Nancy*'. Here's the general pattern:

COPULA: <INSTANCE> is a <CONCEPT>

3.2 Finding Patterns

Having defined these patterns, one could now try to recognize these patterns in a corpus and propose the corresponding instance-concept relationships. However, it is well known that the above patterns are rare and thus one will need a sufficiently big corpus to find a significant number of matches.

Thus, PANKOW resorts to the biggest corpus available: the World Wide Web. In fact, several researchers have shown that using the Web as a corpus is an effective way of addressing the typical data sparseness problem one encounters when working with corpora (compare [14], [21], [23], [25]). Actually, we subscribe to the principal idea by Markert *et al.* [23] of exploiting the Google$^{\text{TM}}$ API. As in their approach, rather than actually downloading web pages for further processing, we just take the number of Web Pages in which a certain pattern appears as an indicator for the strength of the pattern.

Given a candidate proper noun that we want to tag or annotate with the appropriate concept, we instantiate the above patterns with each concept from the given ontology into hypothesis phrases. For each hypothesis phrase, we query the Google$^{\text{TM}}$ API for the number of documents that contain it. The function 'count(i,c,p)' models this query.

$$\text{count} : I \times C \times P \rightarrow \mathbb{N} \tag{1}$$

Thereby, i, c, p are typed variables and short for <INSTANCE>, <CONCEPT> and a pattern. Correspondingly, I, C and P stand for the set of all candidate proper nouns, all concepts from a given ontology and all patterns, respectively.

3.3 Categorizing Candidate Noun Phrases

We have explored three versions for determining the best categorization.

1. Baseline: The simplest version just adds all the numbers of documents with hits for all hypothesis phrases resulting from one <INSTANCE>/<CONCEPT> pair.

   $$\text{count}_b(i,c) := \sum_{p \in P} \text{count}(i,c,p) \tag{2}$$

   This baseline count_b proved to be effective and empirical results presented subsequently will report on this method.

2. Linear Weighting: As it became clear that the different patterns do not indicate the same strength for a <INSTANCE>/ <CONCEPT> pair, we have tried linear weighting of the indicators:

   $$\text{count}_{\vec{w}}(i,c) := \sum_{p \in P} w_p \,\text{count}(i,c,p) \tag{3}$$

   The method to linearly weight the contribution of each pattern is described in Section 4.3.2. This method $\text{count}_{\vec{w}}$, however, has not proved beneficial when compared against count_b so far.

3. Interactive Selection: In an annotation scenario, it is not always necessary to uniquely categorize a candidate proper

noun. Rather it is very easy and effective to present to the manual annotator the top ranked <INSTANCE>/ <CONCEPT> pairs and let him decide according to the actual context. This is currently implemented in CREAM/ OntoMat based on the validity indicated by count$_b$ (also cf. Section 5).

In the first two approaches, one may return the set of pairs R_x ($x \in \{b, \vec{w}\}$) where for a given $i \in I$, $c \in C$ maximizes the strength as aggregated from the individual patterns as the result of PANKOW:

$$R_x := \{(i, c_i) | i \in I, c_i := \text{argmax}_{c \in C} \text{count}_x(i, c)\} \tag{4}$$

and in the last approach, we return the best n matches for each proper noun resulting in R_x^n ($x \in \{b, \vec{w}\}, n \in \mathbb{N}$)[5]:

$$\begin{aligned} R_x^n := \{(i, c_i) | i \in I : c_{i,j} \in C \wedge \{c_{i,1} \ldots c_{i,|C|}\} = C \\ \text{count}_x(i, c_{i,1}) \geq \text{count}_x(i, c_{i,2}) \geq \ldots \geq \text{count}_x(i, c_{i,|C|}) \wedge \\ c_i = c_{i,1} \vee \ldots \vee c_i = c_{i,n}\} \end{aligned} \tag{5}$$

For our evaluation we will use a characterization that does not accept classification of every candidate proper noun, such as R_x does, but only of those that appear strong enough. Thus, we introduce a threshold θ as follows:

$$\begin{aligned} R_{x,\theta} := \{(i, c_i) | i \in I, c_i := \text{argmax}_{c \in C} \text{count}_x(i, c) \wedge \\ \text{count}_x(i, c_i) \geq \theta\} \end{aligned} \tag{6}$$

4. EVALUATION

4.1 Test Set

We have 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 ([27]). The original ontology consisted of 1043 concepts, while the pruned one consisted of 682. The subjects were told to annotate proper nouns 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 categorizations and subject B produced 392. There were 277 proper nouns (referred to by I in the following; $|I| = 277$) that were annotated by both subjects. For these 277 proper nouns, they used 59 different concepts (henceforth constituting our set of concepts C). The categorial agreement on these 277 proper nouns as measured by the Kappa statistic was 63.50% (cf. [5]), which allows to conclude that the classification task is overall well defined. In what follows, we will only consider the proper nouns from I.

4.2 Evaluation Measures

To evaluate our approach, we compare the answers of our system ($R_{b,\theta}$) with the following two reference standards:

- $Standard_A := \{(i, c) |$ for each $i \in I$ the categorization $c \in C$ produced by subject A$\}$
- $Standard_B := \{(i, c) |$ for each $i \in I$ the categorization $c \in C$ produced by subject B$\}$

As evaluation measures, we use the well-known Prec(ision), Rec(all)=Acc(uracy) and F_1-Measures to evaluate our system against $Standard_A$ and $Standard_B$. Prec, Rec and F_1 are defined as follows (for $y \in \{A, B\}$, the two standards):

$$Prec_y = \frac{|\text{correct answers}|}{|\text{total answers}|} = \frac{|R_{b,\theta} \cap Standard_y|}{|R_{b,\theta}|} \tag{7}$$

$$\begin{aligned} Acc_y = Rec_y &= \frac{|\text{correct answers}|}{|\text{answers in reference standard}|} \\ &= \frac{|R_{b,\theta} \cap Standard_y|}{|I|} \end{aligned} \tag{8}$$

Note that recall does not equal accuracy in general, but in the way we defined the classification task, the two are synonym.

$$F_{1,y} = \frac{2 * Prec_y * Rec_y}{Prec_y + Rec_y} \tag{9}$$

Furthermore, in our evaluations we will always average the results for both annotators as given by the following formulas:

$$Prec_{avg} = \frac{Prec_A + Prec_B}{2} \tag{10}$$

$$Acc_{avg} = R_{avg} = \frac{Rec_A + Rec_B}{2} \tag{11}$$

$$F_{1,avg} = \frac{F_{1,A} + F_{1,B}}{2} \tag{12}$$

To get an upper bound for the task, we also calculated the F_1-Measure of $Standard_A$ measured against $Standard_B$ and the other way round and got F_1=62.09% as average.

4.3 Results

For each instance out of the 277 common to $Standard_A$ and $Standard_B$ we have instantiated the 10 patterns[6] described in Section 3 for each concept in the ontology thus resulting in 163,430 (277 x 59 x 10) queries to the Google™ API.

4.3.1 Baseline Experiment

As a baseline experiment we evaluated the results $R_{b,\theta}$ for different $\theta \in \mathbb{N}$. Table 1 gives the 60 categorizations of proper nouns $i \in I$ with highest scores count$_b(i, c)$. Though some of the categorizations were spurious, it became clear that in general the performance of the approach is very good, especially considering that *no effort at all* was invested in defining classification rules or giving training examples.

Figure 2 shows the precision, accuracy (recall) and F_1-Measure values for different thresholds θ averaged over both reference standards: $Standard_A$ and $Standard_B$. Obviously, the precision increases roughly proportionally to the threshold θ, while the accuracy (recall) and F_1-Measure values decrease. We notice that the precision is 50% around threshold $\theta = 64900$ and drops to 0 at threshold $\theta = 65000$. The reason for this is that at threshold 64900 our system produces two answers, i.e., instance(Atlantic,city) and instance(Bahamas,island) (compare Table 1), of which only one is correct. At threshold 65000 the only answer which remains is instance(Atlantic,city), which is wrong according to our annotators. Figure 3 shows the above values for the threshold interval [0..1000]. Interestingly, it can be observed that Prec=Rec=F_1 at $\theta = 0$. The best $F_{1,avg}$-Measure was 28.24% at a threshold of $\theta = 60$ and the best Accuracy Acc_{avg} (R_{avg}) was 24.9% at a threshold of $\theta = 0$.

[5] Obviously, $R_x^1 = R_x$.

[6] The patterns HEARST3 and HEARST4 were actually transformed into two patterns each.

Instance (i)	Concept (c)	count$_b(i,c)$
Atlantic	city	1520837
Bahamas	island	649166
USA	country	582275
Connecticut	state	302814
Caribbean	sea	227279
Mediterranean	sea	212284
South Africa	town	178146
Canada	country	176783
Guatemala	city	174439
Africa	region	131063
Australia	country	128607
France	country	125863
Germany	country	124421
Easter	island	96585
St Lawrence	river	65095
Commonwealth	state	49692
New Zealand	island	40711
Adriatic	sea	39726
Netherlands	country	37926
St John	church	34021
Belgium	country	33847
San Juan	island	31994
Mayotte	island	31540
EU	country	28035
UNESCO	organization	27739
Austria	group	24266
Greece	island	23021
Malawi	lake	21081
Israel	country	19732
Perth	street	17880
Luxembourg	city	16393
Nigeria	state	15650
St Croix	river	14952
Nakuru	lake	14840
Kenya	country	14382
Benin	city	14126
Cape Town	city	13768
St Thomas	church	13554
Niger	river	13091
Christmas Day	day	12088
Ghana	country	10398
Crete	island	9902
Antarctic	continent	9270
Zimbabwe	country	9224
Central America	region	8863
Reykjavik	island	8381
Greenland	sea	8043
Cow	town	7964
Expo	area	7481
Ibiza	island	6788
Albania	country	6327
Honduras	country	6143
Iceland	country	6135
Nicaragua	country	5801
Yugoslavia	country	5677
El Salvador	country	5154
Senegal	river	5139
Mallorca	island	4859
Nairobi	city	4725
Cameroon	country	4611
Rust	park	4541

Table 1: Top 60 Instance-Concept Relations

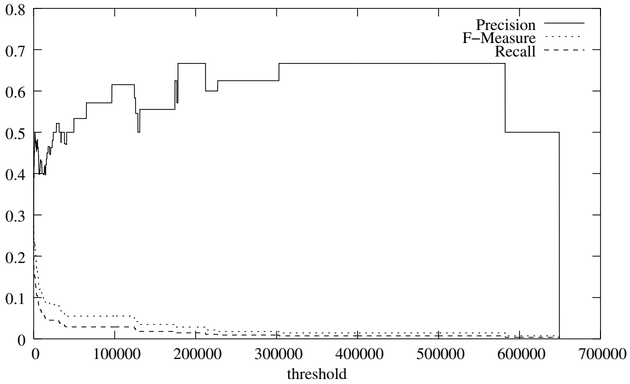


Figure 2: Precision, F-Measure and Accuracy/Recall for $R_{b,\theta}$

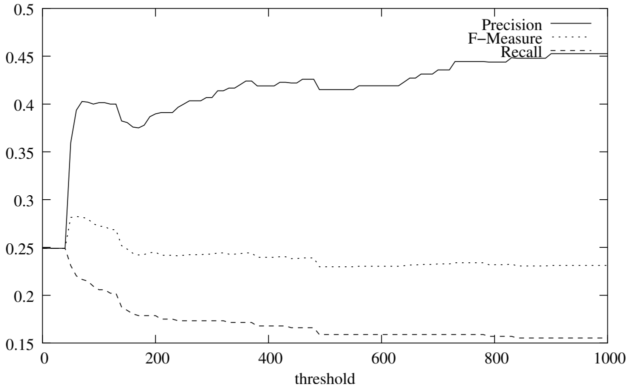


Figure 3: Precision, F-Measure and Accuracy/Recall for $R_{b,\theta}$ zoomed into interval [0..1000]

4.3.2 Weighting the Patterns

As a second experiment, we tried to find out if it is useful to treat the contribution of each pattern separately and weight the patterns against each other. For this purpose, we randomly selected 500 more Web Pages from *Lonely Planet*[7]. From these pages, we extracted potential instance-concept candidates, i.e., we matched the patterns described in section 3 against the texts. We used a part-of-speech (POS) tagger (cf., e.g., [4, 26]) to find such candidate proper nouns. A part-of-speech tagger assigns the correct syntactic category, i.e., adjective, common noun, proper noun to words. Typically, they exploit the surrounding context of a word as features and some learning algorithm to induce corresponding tagging rules. On the basis of the POS-tagger output we interpret sequences of words tagged as PN (proper nouns) as instances and the head of noun phrases as concepts.

We then presented these instance-concept pairs to three different subjects for validation. They had the possibility of validating the relationship, adding the concept name to the instance, rejecting the relationship or expressing their doubt. The possibility of adding the concept name is important when judging a suggestion such as that *Lenin* is an instance of a *museum*. In this case, the users could decide that the suggestion of the system is not totally wrong and correct the suggestion by specifying that *Lenin museum* is the actual instance of a *museum*.

[7] http://www.lonelyplanet.com/destinations

Pattern	Suggested	Annotator1	Annotator 2	Annotator 3	Average
HEARST1	2	40.00%	40.00%	60.00%	46.66%
DEFINITE1	19	21.05%	36.84%	36.84%	31.56%
DEFINITE2	74	91.36%	93.83%	96.30%	93.83%
APPOSITION	28	56.00%	62.00%	62.00%	60.00%
COPULA	22	66.67%	66.67%	63.64%	65.66%
ALL	188	69.15%	73.40%	74.47%	72.34%

Table 2: Accuracy of each of the patterns

Pattern	Relative Weight
HEARST1-4	5
DEFINITE1	3
DEFINITE2	9
APPOSITION	6
COPULA	7

Table 3: Relative weights of the patterns

Table 2 gives the accuracy for all the patterns based on the answers of the human subjects to the suggestions of the system. Unfortunately, no HEARST2, HEARST3 or HEARST4 instances were found in the texts, which shows that they are actually the ones which occur most rarely.

The above task can be seen as a classification task of the suggested instance-concept relationships into the four categories: *correct*, *add concept*, *wrong* and *doubt*. Thus, we can measure the categorial agreement between the three annotators as given by the Kappa statistic ([5]). In fact, when computing the average of the pairwise agreement between the annotators, we yield a Kappa value of K=66.19%. Thus the agreement seems quite reasonable and according to [5] is almost in a range from which ‘tentative conclusions’ can be drawn. This in turn means that our task is well defined.

The results itself show that in general the accuracy of the patterns is relatively good, i.e., almost 3/4 of the suggested instance-concept relations are correct. It also shows that the Hearst patterns are extremely rare. It is also interesting to notice that the *DEFINITE1* and *DEFINITE2* patterns, though they share the same rationale, have a completely different performance in terms of accuracy. Finally - and most importantly - the results show that the performance of each of the patterns in terms of accuracy is very different such that there is an actual need of weighting the contribution of each pattern. As a first approximation of setting the weights of the patterns to maximize the overall accuracy of the approach, we decided to weight the patterns relatively to each other proportionally to their accuracy. In particular we used the relative weights in Table 3. However, we found out that weighting the patterns in this linear fashion makes the results actually worse. In fact, the best F-Measure was $F_{1,avg} = 24.54\%$ ($t = 290$) and the best accuracy was $Acc_{avg} = 21.48\%$ ($t = 0$). As a further experiment we also tried to find optimal weights by training a neural network as well as other classifiers. However, due to the lack of a representative number of (positive) training examples, the model learned by the classifiers was worse than our baseline.

4.3.3 Interactive Selection

When using the interactive selection variant, i.e., $R^5_{b,0}$, if one of the top 5 answers of our system coincides with the one given by the annotator, we count it as a correct answer. Thus, we obviously get higher F-Measure, Precision and Accuracy (Recall) values. They are depicted in comparison to the baseline in Figure 4. The best accuracy here is Acc_{avg}=49.56%. This means in practice that for almost half of the instances in a web page, we provide the user already with the correct answer, thus notably reducing the annotation time and cost.

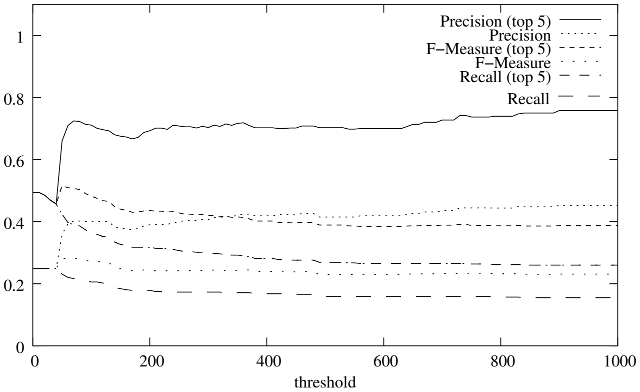


Figure 4: Precision, F-Measure and Accuracy/Recall for $R^5_{b,0}$ compared to $R_{b,\theta}$ zoomed into interval [0..1000]

4.4 Discussion

The results of the experiment described above are certainly very encouraging. As Table 1 shows, the overall results of our automatic classification seem quite reasonable. Some instance-concept relationships are certainly spurious, such as, for example, that *South Africa* is a *town*. In fact, the second best ranked category of our approach for *South Africa* is the correct one, i.e., *country*. Thus, a semi-automatic use of our approach in which the users are asked to select one of the highest ranked categories increases considerably the performance of our approach (compare section 4.3.3).

From a quantitative point of view, the best Accuracy of 24.9% is comparable to state-of-the-art systems performing a similar classification task, especially given the fact that our approach is unsupervised and does not require text preprocessing methods (see Section 6). The performance of our system is still far away from the human performance on the task ($F_1 = 62.09\%$), but it is also quite away from a random decision procedure. Thus, the results of our approach seem very promising. In future experiments we will verify if these results are scalable to a larger set of concepts such as the ca. 1200 considered by Alfonseca and Manandhar ([2]).

5. INTEGRATION INTO CREAM

We have integrated PANKOW into the CREAM framework [16] extending the CREAM implementation OntoMat by a plugin. The plugin has access to the ontology structure and to the document management system of OntoMat. The plugin utilizes the Google^{TM} API to access its web service.

The PANKOW plugin implements the process described in Section 2 starting with the scanning for the candidate proper nouns by using a POS tagger as described in Section 3. We experimented with two POS taggers: One was QTag[8] and the other was Tree-Tagger ([26]). The advantage of QTag is that it is implemented in Java and therefore better to integrate. Whereas, Tree-Tagger produces somewhat better results in our experiments.

In addition, we use a heuristic to get a higher precision for the candidate recognition and therefore to reduce the amount of queries. The heuristic considers the intersection of the POS tagger categorization with the simple capitalized-words approach which consists in interpreting sequences of capitalized words as proper noun candidates[9]. For the capitalized words approach we consider only words that do not follow a period. Given the example of the lonely planet web page about Nigeria[10], the POS tagger proposes proper nouns such as “Guinea”, “Niger”, “Cameroon”, “Benin”, and “Nigeria”. For this concrete example, the capitalized words approach proposes basically the same proper nouns as the POS tagger. However, in general the capitalized word heuristic will reduce tagging errors produced by the POS tagger. While our heuristic approach is practical, it has some problems with compound words such as “Côte d’Ivoire” and might need some fine-tuning.

OntoMat supports two modes of interaction with PANKOW: *(i)*, fully automatic annotation and, *(ii)*, interactive semi-automatic annotation. In the fully automatic mode, all categorizations with strength above a user-defined θ, viz. $R_{b,\theta}$, are used to annotate the Web content. In the interactive mode, the system proposes the top five concepts to the user for each instance candidate, i.e. R_b^5. Then, the user can disambiguate and resolve ambiguities.

The screenshot in Figure 5 shows the user interface. In the lower left corner of the screenshot you can see the progress dialog for the Google^{TM} queries. The dialog shows the extracted candidate proper nouns and logs the query results for the hypothesis phrases. Also shown is the interactive dialog for disambiguation, e.g. the choice to assign “Niger” as an instance to one of the concepts “river”, “state”, “coast”, “country” or “region”. The number in the brackets behind each concept name gives the number of Web hits.

6. RELATED WORK

We have presented a novel paradigm: *the self-annotating Web* as well as an original method PANKOW which makes use of globally available structures as well as statistical information to annotate Web resources. Though there are initial blueprints for this paradigm, to our knowledge there has been no explicit formulation of this paradigm as well as a concrete application of it as presented in this paper before.

On the other hand, there is quite a lot of work related to the use of linguistic patterns to discover certain ontological relations from text. Hearst [19] and Charniak [6], for example, make use of a related approach to discover taxonomic and *part-of* relations from text, respectively. The accuracy of the *isa*-relations learned by Hearst is 61/106 (57.55%) when measured against WordNet as gold standard. The accuracy of the *part-of* relations is 55% measured against the intuitions of human subjects.

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. The aim of the Named Entity Task as defined in the MUC conference series ([20]) is to assign the categories *ORGANIZATION*, *PERSON* and *LOCATION*. State-of-the-art approaches typically achieve F-Measures over 90% — however that challenge of categorizing into 3 classes was quite modest when compared against the challenge of categorizing into 59 classes.

Other researches have considered this harder task such as Hahn and Schnattinger [15], Alfonseca and Manandhar [2] or Fleischman and Hovy [11].

Hahn and Schnattinger [15] 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 [15]) in terms of accuracy can be found in Table 4. 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. On the one hand they tackle categorization into an even larger number of concepts than we do and hence our task would be easier. On the other hand they evaluate their approach under clean room conditions as they assume accurately identified syntactic and semantic relationships and an elaborate ontology structure, while we our evaluation is based on very noisy input — rendering our task harder than theirs. Nevertheless, as a vague indication, Table 4 compares our approaches (among others).

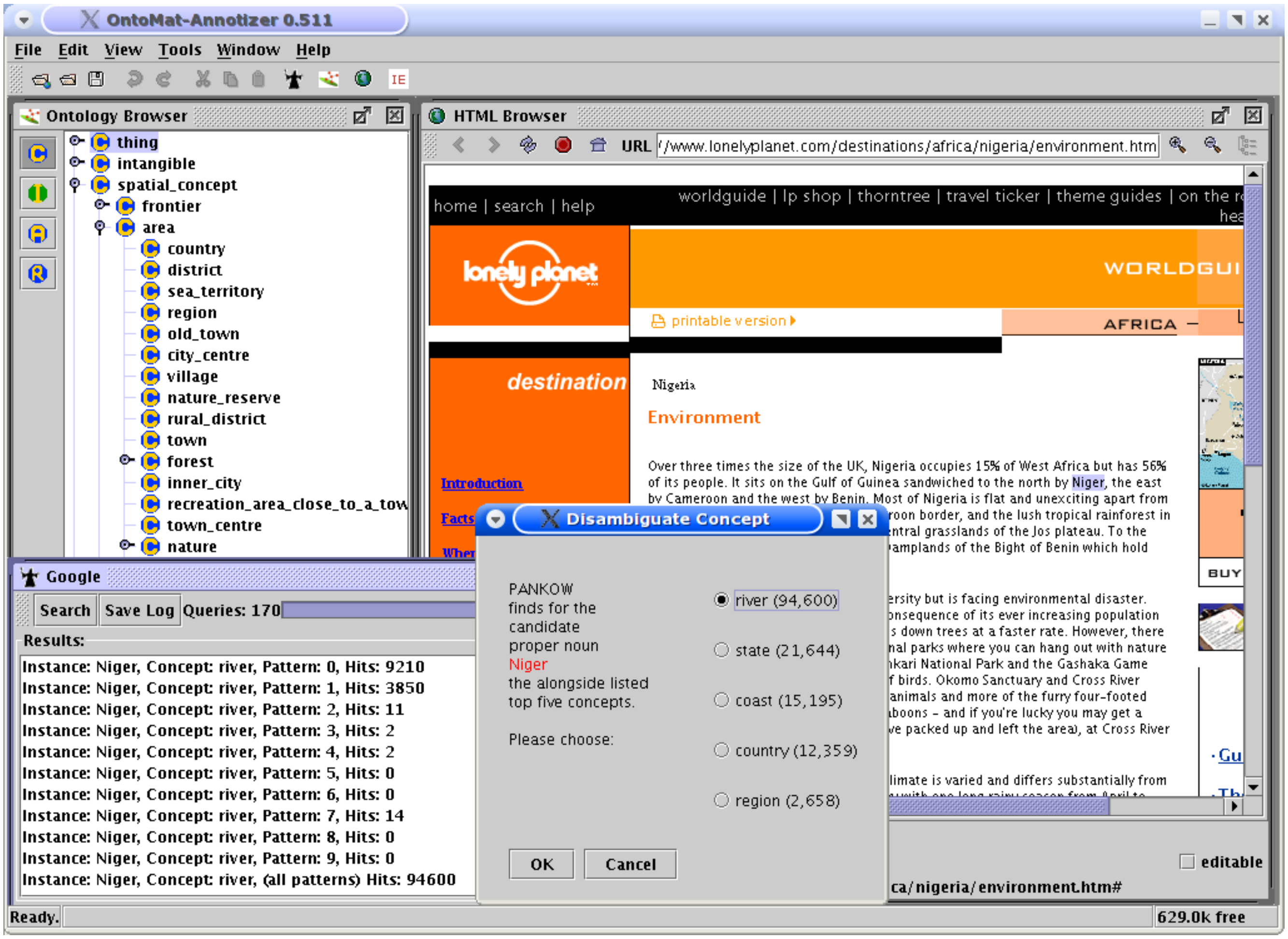
Alfonseca and Manandhar [2] have also addressed the problem of assigning the correct ontological class to unknown words. Their system is based on the distributional hypothesis, i.e., that words are similar to the extent to which they share linguistic contexts. In this line, they adopt a vector-space 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 results of their approach are also given in Table 4. However, it is important to mention that it is not clear from their paper if they are actually evaluating their system on the 1200 synsets/concepts or only on a smaller subset of them.

Fleischmann and Hovy [11] 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. [15] 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 away from the number of categories we consider here.

[8] `http://web.bham.ac.uk/o.mason/software/tagger/index.html`

[9] This heuristic works especially well for English, where typically only proper nouns appear capitalized.

[10] `http://www.lonelyplanet.com/destinations/africa/nigeria/environment.htm`



URL '/www.lonelyplanet.com/destinations/africa/nigeria/environment.htm

Nigeria

Environment

Over three times the size of the UK, Nigeria occupies 15% of West Africa but has 56% of its people. It sits on the Gulf of Guinea sandwiched to the north by Niger, the east by Cameroon and the west by Benin. Most of Nigeria is flat and unexciting apart from
roon border, and the lush tropical rainforest in
ntral grasslands of the Jos plateau. To the
amplands of the Bight of Benin which hold

ersity but is facing environmental disaster.
onsequence of its ever increasing population
s down trees at a faster rate. However, there
nal parks where you can hang out with nature
nkari National Park and the Gashaka Game
f birds. Okomo Sanctuary and Cross River
animals and more of the furry four-footed
aboons – and if you're lucky you may get a
ve packed up and left the area), at Cross River

limate is varied and differs substantially from

ca/nigeria/environment.htm#

```
Instance: Niger, Concept: river, Pattern: 0, Hits: 9210
Instance: Niger, Concept: river, Pattern: 1, Hits: 3850
Instance: Niger, Concept: river, Pattern: 2, Hits: 11
Instance: Niger, Concept: river, Pattern: 3, Hits: 2
Instance: Niger, Concept: river, Pattern: 4, Hits: 2
Instance: Niger, Concept: river, Pattern: 5, Hits: 0
Instance: Niger, Concept: river, Pattern: 6, Hits: 0
Instance: Niger, Concept: river, Pattern: 7, Hits: 14
Instance: Niger, Concept: river, Pattern: 8, Hits: 0
Instance: Niger, Concept: river, Pattern: 9, Hits: 0
Instance: Niger, Concept: river, (all patterns) Hits: 94600
```

PANKOW finds for the candidate proper noun Niger the alongside listed top five concepts.

Please choose:

● river (94,600)
○ state (21,644)
○ coast (15,195)
○ country (12,359)
○ region (2,658)

Figure 5: Screenshot of CREAM with PANKOW plugin in interactive mode

In [17] we proposed a semi-automatic approach to discovering instances of a concept by using a machine-learning based information extraction system (viz. Amilcare [7]). However, this approach (as well as others, e.g. [28]) presupposes a certain amount of manually annotated pages on which the system can be trained. With the approach presented here we certainly overcome the burden of manual training of the system.

7. CONCLUSION

We have described PANKOW, a novel approach towards the *Self-annotating Web*. It overcomes the burden of laborious manual annotation and it does not require the manual definition of an information extraction system or its training based on manually provided examples. It uses the implicit wisdom contained in the Web to propose annotations derived from counting Google^{TM} hits of instantiated linguistic patterns. The results produced are comparable to state-of-the-art systems, whereas our approach is comparatively simple, effortless and intuitive to use to annotate the Web.

While we consider PANKOW as a valid step towards the self-annotating Web, we are well-aware that effectiveness, efficiency, and range of PANKOW needs to and can be improved.

With regard to effectiveness, we have mentioned in the beginning that [10, 12] used the Web structure itself to determine a focus for harvesting data. In this line, by determining such a focus we could have a more domain-specific disambiguation than in our current approach. Such a focus could, for example, be determined by crawling only similar documents from the Web as for example in the approach of Agirre et al. ([1]). For instance, our annotators tagged 'Niger' as a country, while PANKOW found the other meaning that 'Niger' has, viz., it also refers to the river 'Niger'.

With regard to efficiency, we thank Google^{TM} for their help and support with their Web service API. However, the self-annotating Web is not possible with the kind of implementation that we provided. The reason is that we have issued an extremely large number of queries against the Google^{TM} API — to an extent that would not scale towards thousands or even millions of Web pages in a reasonable time-scale. However, we envision an optimized indexing scheme (e.g., normalizing the various forms of '*to be*' in order to recognize '*George Washington was a man*') and API that would reduce this number to acceptable load levels. Also, an interesting direction for further research would be to learn the weights of the different patterns by machine-learning techniques. Furthermore, in order to reduce the amount of queries sent to the Google Web service API, a more intelligent strategy should be devised, which takes into account the ontological hierarchy between concepts.

With regard to range, we have only covered the relationship between instances and their concepts, but not other relationships be-

System	No. Concepts	Preprocessing	Accuracy
MUC	3	various	>90%
Fleischman et al.	8	N-gram frequency extraction	70.4%
PANKOW ($R_{b,0}$)	59	none	24.9%
PANKOW ($R_{\vec{w}}$)	59	none	21.48%
PANKOW ($R^{5}_{b,0}$)	59	none	49.46%
Hahn et al. (Baseline)	196	perfect syntactic and semantic analysis required	21%
Hahn et al. (TH)	196	perfect syntactic and semantic analysis	26%
Hahn et al. (CB)	196	perfect syntactic and semantic analysis	31%
Alfonseca et al.	1200 (?)	syntactic analysis	28.26%

Table 4: Comparison of results

tween instances, such as *is_located_in(Eiffel Tower,Paris)*. Our first step in this direction will be the tighter integration of PANKOW with Amilcare [7], such that instance data from PANKOW will be used to train Amilcare as has been done for Armadillo [8]. Overall, however, this remains extremely challenging work for the future.

Acknowledgments. We would like to thank Google for their support with their Web service API. Thanks to Kai Kühn for help with implementing Pankow and integrating it into OntoMat. Further, we would also like to acknowledge Ursula Sciesinski and Kai Kühn for annotating the Lonely Planet Web pages. Thanks also to our colleagues for their support as well as to the reviewers of the WWW conference for valuable comments on our paper. Research for this paper has been funded by the projects DARPA DAML OntoAgents (01IN901C0) and IST-Dot.Kom (grant IST-2001-34038). The Dot.Kom project (http://www.dot-kom.org) is sponsored by the EC as part of the framework V.

8. REFERENCES

[1] E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching Very Large Ontologies using the WWW. In *Proceedings of the First Workshop on Ontology Learning OL'2000 Berlin, Germany, August 25, 2000*, 2000. Held in conjunction with the 14th European Conference on Artificial Intelligence ECAI'2000, Berlin, Germany.

[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] D. Appelt, J. Hobbs, J. Bear, D. Israel, M. Kameyama, and M. Tyson. FASTUS: a finite state processor for information extraction from real world text. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI)*, 1993.

[4] Eric Brill. Some advances in transformation-based part of speech tagging. In *National Conference on Artificial Intelligence*, pages 722–727, 1994.

[5] J. Carletta. Assessing 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] F. Ciravegna. Adaptive Information Extraction from Text by Rule Induction and Generalisation. In Bernhard Nebel, editor, *Proceedings of the Seventeenth International Conference on Artificial Intelligence (IJCAI-01)*, pages 1251–1256, San Francisco, CA, August 2001. Morgan Kaufmann Publishers, Inc.

[8] F. Ciravegna, A. Dingli, D. Guthrie, and Y. Wilks. Integrating Information to Bootstrap Information Extraction from Web Sites. In *IJCAI 2003 Workshop on Information Integration on the Web, workshop in conjunction with the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico, August, 9-15*, pages 9–14, 2003.

[9] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the Twelfth International Conference on World Wide Web*, pages 178–186. ACM Press, 2003.

[10] G.W. Flake, S. Lawrence, C.L. Giles, and F.M. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3):66 –70, March 2002.

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

[12] Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, and Gary W. Flake. Using web structure for classifying and describing web pages. In *Proceedings of the Eleventh International Conference on World Wide Web*, pages 562–569. ACM Press, 2002.

[13] Googlism, 2003. http://www.googlism.com.

[14] G. Grefenstette. The WWW as a resource for example-based MT tasks. In *Proceedings of ASLIB'99 Translating and the Computer 21*, 1999.

[15] 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*, pages 524–531, 1998.

[16] S. Handschuh and S. Staab. Authoring and annotation of web pages in CREAM. In *Proceedings of the 11th International World Wide Web Conference, WWW 2002, Honolulu, Hawaii, May 7-11, 2002*, pages 462–473. ACM Press, 2002.

[17] S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM — Semi-automatic CREAtion of Metadata. In *Proceedings of EKAW 2002*, LNCS, pages 358–372, 2002.

[18] S. Handschuh, S. Staab, and A. Maedche. CREAM — Creating relational metadata with a component-based, ontology-driven annotation framework. In *Proceedings of K-Cap 2001*, pages 76–83. ACM Press, 2001.

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

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

[21] Frank Keller, Maria Lapata, and O. Ourioupina. Using the web to overcome data sparseness. In *Proceedings of EMNLP-02*, pages 230–237, 2002.

[22] A. Mädche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, March/April 2001.

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

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

[25] P. Resnik and N. Smith. The web as a parallel corpus. *Computational Linguistics*, 29(3), 2003.

[26] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, 1994.

[27] 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.

[28] M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology Driven Semi-automatic and Automatic Support for Semantic Markup. In *Proceedings of EKAW 2002*, LNCS 2473, pages 379–391. Springer, 2002.