EAGER: Extending Automatically Gazetteers for Entity Recognition

Omer Gunes, Christian Schallhart, Tim Furche
Department of Computer Science,
Oxford University, Oxford OX1 3QD
firstname.lastname@cs.ox.ac.uk

Jens Lehmann, Axel Ngonga
Institute of Computer Science,
University of Leipzig, 04103 Leipzig
lastname@informatik.uni-leipzig.de

Abstract

Key to named entity recognition, the manual gazetteering of entity lists is a costly, error-prone process that often yields results that are incomplete and suffer from sampling bias. Exploiting current sources of structured information, we propose a novel method for extending minimal seed lists into complete gazetteers. Like previous approaches, we value WIKIPEDIA as a huge, well-curated, and relatively unbiased source of entities. However, in contrast to previous work, we exploit not only its content, but also its structure, as exposed in DBPEDIA. We extend gazetteers through Wikipedia categories, carefully limiting the impact of noisy categorizations. The resulting gazetteers easily outperform previous approaches on named entity recognition.

1 Introduction

Automatically learning gazetteers with minimal supervision is a long standing problem in named entity recognition.

We propose EAGER as a novel approach to extending automatically gazetteers for entity recognition, utilizing DBPEDIA (Bizer et al., 2009) rather than WIKIPEDIA. DBPEDIA serves as a much better foundation than WIKIPEDIA, because all the information used in previous approaches (and much more) is already provided as a structured database of facts and articles. The extraction is more robust and complete than ad-hoc methods and maintained by a large community. E.g., navigating the category hierarchy is much easier and reliable with DBPEDIA.

To summarize, EAGER's main contributions are

(1) A *novel gazetteer expansion algorithm* that adds new entities from DBPEDIA. EAGER adds entities that have several categories in common with the seed terms, addressing noisy categorizations through a sophisticated *category pruning technique.*

(2) EAGER also extracts categories from DBPEDIA abstracts using *dependency analysis*. Finally, EAGER extracts plural forms and synonyms from *redirect information*.

(3) For *entity recognition*, we integrate the gazetteer with a simple, but effective machine learning classifier, and *experimentally* show that the extended gazetteers improve the F_1 score between 7% and 12% over our baseline approach and outperform (Zhang and Iria, 2009) on all learned concepts (subject, location, temporal).

2 Related Work

We divide the related work in automatic gazetteer population into three groups: **(1)** *Machine learning* approaches **(2)** *Pattern driven* approaches Finally, like our own work, **(3)** *knowledge driven* approaches

Knowledge Driven. In any case, machine learning and pattern driven approaches extract their terms from unstructured sources – despite the fact that large, general knowledge bases became available in the last years. One of the first knowledge-driven methods (Magnini et al., 2002) employed WORDNET to identify trigger words and candidate

Proceedings of the 3rd Workshop on the People's Web Meets NLP, ACL 2012, pages 29–33,
Jeju, Republic of Korea, 8-14 July 2012.

gazetteer terms with its word-class and -instance relations. As WORDNET covers domain specific vocabularies only to a limited extent, this approach is also limited in its general applicability.

In (Toral and Muñoz, 2006), gazetteers are built from the noun phrases in the first sentences of WIKIPEDIA articles by mapping these phrases to WORDNET and adding further terms found along the hypernymy relations. The approach presented in (Kazama and Torisawa, 2007; Kazama and Torisawa, 2008) relies solely on WIKIPEDIA, producing gazetteers without explicitly named concepts, arguing that consistent but anonymous labels are still useful.

Most closely related to our own work, the authors of (Zhang and Iria, 2009) build an approach solely on WIKIPEDIA which does not only exploit the article text but also analyzes the structural elements of WIKIPEDIA:

3 Automatically Extending Gazetteer Lists

3.1 Extraction Algorithm: Overview

Algorithm 1 shows an outline of the gazetteer expansion algorithm used in EAGER. To extend an initial seed set $\mathscr{S}$ EAGER proceeds, roughly, in three steps: First, it identifies DBPEDIA articles for seed entities and extracts implicit category and synonym information from abstracts and redirect information (Lines 1–11). Second, it finds additional categories from the DBPEDIA category hierarchy (Lines 12–20). Finally, it uses the categories from the first two steps to extract additional entities (Lines 21–24). In the following, we consider the three steps separately.

Algorithm 1: GazetteerExtension($\mathscr{S}$)

```
1  foreach seed entity e ∈ S do
2      find article a for e in DBPEDIA;
3      Articles(e) ← a;
4  G ← ∅; P ← ∅;
5  foreach entity e, article a = Articles(e) do
6      foreach sentence s ∈ a.Abstract do
7          D_s ← dependencies in s;
8          add all t : nsubj(e,t) ∈ D_s to P;
9          add all t : nsubj(e,t'), conj(t',t) ∈ D_s to P;
10     foreach article a' ∈ a.Redirects do
11         add all labels of a to G;
12     Cats(e) ← Cats(e) ∪ a.Cats;
13 foreach entity e, category c ∈ Cats(e) do
14     Cats(e) ← Cats(e) ∪ CategoryNeighbors(c,k);
15 foreach category c ∈ Cats(e) for some e do
16     Support(c) ← |{e' : c ∈ Cats(e')}|;
17 foreach connected component C in Cats do
18     Support(C) ← Σ_{c∈C} Support(c);
19 MaxCatComp ← C with maximal Support;
20 add all categories in MaxCatComp to P;
21 foreach category c ∈ P do
22     foreach article a with c ∈ a.Cats do
23         if |a.Cats \ P| ≤ θ then
24             add all labels of a to G;
```

3.2 Implicit: Abstract and Redirects

Before EAGER can analyse abstract and redirect information for an article, we need to **find the corresponding DBPEDIA articles** (Lines 1–3) for each seed entry in $\mathscr{S}$. There may be one or more such entry. Here, we observe the first advantage of DBPEDIA's more structured information: DBPEDIA already contains plain text labels such as "Barack Obama" and we can directly query (using the SPARQL endpoint) all articles with a label equal (or starting with) an entity in our seed set. This allows for more precise article matching and avoids complex URL encodings as necessary in previous, WIKIPEDIA-based approaches such as (Kazama and Torisawa, 2007). As (Zhang and Iria, 2009), we reject redirection entries in this step as ambiguous.

With the articles identified, we can proceed to extract category information from the abstracts and new entities from the redirect information. In the **dependency analysis of article abstracts** (Lines 6–9), we aim to extract category (or, more generally, hypernym) information from the abstracts of articles on the ssed list. We perform a standard dependency analysis on the sentences of the abstract and return all nouns that stand in `nsubj` relation to a seed entity or (directly or indirectly) in `conj` (correlative conjunction) relation to a noun that stands in `nsubj` relation to a seed entity. This allows us to extract, e.g., both "general" and "statesman" as categories from a sentence such as "Julius Caesar was a Roman general and statesman". This analysis is inspired by (Zhang and Iria, 2009), but performed on the entire abstract which is clearly dis-

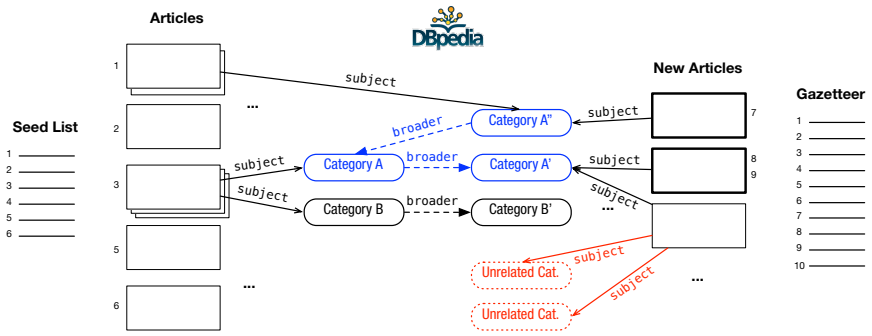

Figure 1: EAGER Gazetteer Extension Algorithm

tinguished in WIKIPEDIA. This contrasts to (Zhang and Iria, 2009), where this is applied only to the first sentence (as WIKIPEDIA does not directly provide a concept of "abstract"). All categories thus obtained are added to $\mathscr{P}$ and will be used in Section 3.4 to generate additional entities.

Finally, we are interested in **redirection information** (Lines 10–11) about an article for a seed entity as that provides such with synonyms, plural forms, different spellings, etc. Fortunately, DBPEDIA provides this information directly by means of the `dbpedia-owl:wikiPageRedirects` property. The labels of all redirect articles with this property pointing to a seed entity articles are directly added to the Gazetteer.

3.3 Explicit: Category Graph

In addition to categories from the abstract analysis, we also use the **category graph** of DBPEDIA. It has been previously observed, (Zhang and Iria, 2009) and (Strube and Ponzetto, 2006), that the category graph of poor quality. DBPEDIA improves little on that fact. However, EAGER uses a sophisticated analysis of categories related to seed entities that allows us to prune most of the noise in the category graph. Biased towards precision over recall, Section 4 shows that combined with the category extraction from abstracts it provides a significantly extended Gazetteer without introducing substantial noise.

The fundamental contribution of EAGER is a category pruning based on finding a connected component in the graph of related categories that is supported by as many different entities from the seed list as possible. Figure 1 illustrates this further: From the articles for the seed entities, we compute (Line 12) the direct categories (via `subject` edges) and associate them to their seed entities e via $\texttt{Cats}(e)$. We extent this set (Lines 13–14) with all categories in the k-neighbourhood (here, we use $k = 3$), i.e., connected via up to k `broader` edges traversed in any direction, again maintaining via $\texttt{Cats}(e)$ which categories are reached from which seed entity e. In the resulting graph of all such categories, we identify the connected component with *maximum support* (Lines 15–19). The support of a component is the sum of the support of its categories. The support of a category c is the number of seed entities with $c \in \texttt{Cats}(e)$. For Figure 1, this yields the category graph of the blue and black categories of the figure. The blue categories form the connected component with maximum support and are thus retained (in $\mathscr{P}$), the black categories are dropped.

3.4 Entities from Categories

Finally, in Lines 21–24, EAGER completes the gazetteer extension by extracting the labels of all articles of categories in $\mathscr{P}$ if they are *sufficiently unambiguous*. An article is called sufficiently unambiguous, if it is categorised only with categories from $\mathscr{P}$ up to a threshold θ (here, set to 5) of non-$\mathscr{P}$ categories. This avoids adding very general entities that tend to have large number of categories in WIKIPEDIA and thus DBPEDIA. The output of Algorithm 3 is the extended gazetteers $\mathscr{G}$.

4 Evaluation

To evaluate the impact of EAGER on entity recognition, we performed a large set of experiments (on the archeology domain). The experiment domains and

	Subject			Location			Temporal		
	P	*R*	F_1	*P*	*R*	F_1	*P*	*R*	F_1
Baseline (B)	69.9	54.1	62.0	76.5	46.1	61.3	86.4	75.4	80.9
B+ Dependency	73.6	64.7	69.0	73.4	69.4	71.4	86.2	85.2	85.7
B+ Category	72.3	64.5	68.5	71.9	70.1	71.0	86.4	85.0	85.8
B+ Redirection	71.6	65.8	68.7	71.2	71.7	71.4	86.32	85.46	85.89
(Zhang and Iria, 2009) full	69.8	66.5	68.1	68.9	75.0	71.8	82.4	83.4	82.9
EAGER *full*	72.1	66.5	69.3	72.0	74.6	73.3	86.8	86.1	86.5

Table 1: EAGER comparison

corpora are described in Section 4.1. Finally, Section 4.2 presents the results of the evaluation, showing the contributions of the different parts of EAGER and comparing it with (Zhang and Iria, 2009), which we outperform for all entity types, in some cases up to 5% in F_1 score.

4.1 Evaluation Setup

In this experiment, we consider entity recognition in the domain of archaeology.

As part of this effort, (Jeffrey et al., 2009) identified three types of entities that are most useful for archaeological research; Subject(SUB), Temporal Terms(TEM), Location (LOC).

In this evaluation, we use the same setup as in (Zhang and Iria, 2009): A corpus of 30 full length UK archaeological reports archived by the Arts and Humanities Data Service (AHDS). The length of the documents varies from 4 to 120 pages. The corpus is inter-annotated by three archaeologists.

4.2 Result

For the evaluation, we perform a 5-fold validation on the above corpus. The evaluate the performance (in terms of precision, recall and F_1 score) for entity recognition of the baseline system as well as the baseline system extended with a gazetteer feature. For the latter, we consider full EAGER as described in Section 3 as well as only the entities derived from dependency analysis of abstracts, from the category graph, and from redirection information. Finally, we also include the performance numbers report in (Zhang and Iria, 2009) for comparison (since we share their evaluation settings).

Table 1 show the results of the comparison: EAGER significantly improves precision and recall over the baseline system and outperforms(Zhang and Iria, 2009) in all cases. Furthermore, the impact of all three types of information (dependencies from abstract, category, redirection) of EAGER individually is quite notable with a slight disadvantage for category information. However, in all cases the combination of all three types as proposed in EAGER shows a significant further increase in performance.

5 Conclusion

At its heart, EAGER is a novel algorithm for extending sets of entities of a specific type with additional entities of that type extracted from DBPEDIA. It is based on a new strategy for pruning the category graph in DBPEDIA (and thus WIKIPEDIA), necessary to address the inherent noise. Our evaluation shows that EAGER can significantly improve the performance of entity recognition and outperforms existing systems in all cases. Unlike previous approaches, our approach makes use of richer content and structural elements of DBpedia.

We believe that EAGER is a strong indicator that DBPEDIA provides a much richer, yet easier to use foundation for NLP tasks in general than WIKIPEDIA.

The extensibility and domain adaptability of our methods still need further investigation. We are currently extending the evaluation to several other domains, including property descriptions in real estate and classified adds. We are also investigating more targeted means of detecting and addressing noise in the category graph.

References

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia – A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3).

Stuart Jeffrey, Julian Richards, Fabio Ciravegna, Stewart Waller, Sam Chapman, and Ziqi Zhang. 2009. The Archaeotools project: Faceted Classification and Natural Language Processing in an Archaeological Context. *Phil. Trans. R. Soc. A*, 367(3):2507–2519.

Jun'ichi Kazama and Kentaro Torisawa. 2007. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL, pages 698–707.

Jun'ichi Kazama and Kentaro Torisawa. 2008. Inducing Gazetteers for Named Entity Recognition by Large-Scale Clustering of Dependency Relations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 407–415.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002. A WordNet-based approach to Named Entities recognition. In *Proceedings of the 2002 workshop on Building and using semantic networks - Volume 11*, SEMANET '02, pages 1–7.

M. Strube and S. P. Ponzetto. 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1419–1424.

Antonio Toral and Rafael Muñoz. 2006. A Proposal to Automatically Build and Maintain Gazetteers for Named Entity Recognition by using Wikipedia. In *Proceedings of the Workshop on New Text – wikis and blogs and other Dynamic text sources*, ECAL'06, pages 56–61.

Ziqi Zhang and José Iria. 2009. A novel approach to automatic gazetteer generation using Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, People's Web '09, pages 1–9.