A Novel Approach to Automatic Gazetteer Generation using Wikipedia

Ziqi Zhang
University of Sheffield, UK
z.zhang@dcs.shef.ac.uk

José Iria
University of Sheffield, UK
j.iria@dcs.shef.ac.uk

Abstract

Gazetteers or entity dictionaries are important knowledge resources for solving a wide range of NLP problems, such as entity extraction. We introduce a novel method to automatically generate gazetteers from seed lists using an external knowledge resource, the Wikipedia. Unlike previous methods, our method exploits the rich content and various structural elements of Wikipedia, and does not rely on language- or domain-specific knowledge. Furthermore, applying the extended gazetteers to an entity extraction task in a scientific domain, we empirically observed a significant improvement in system accuracy when compared with those using seed gazetteers.

1 Introduction

Entity extraction is the task of identifying and classifying atomic text elements into predefined categories such as person names, place names, and organization names. Entity extraction often serves as a fundamental step for complex Natural Language Processing (NLP) applications such as information retrieval, question answering, and machine translation. It has been recognized that in this task, gazetteers, or entity dictionaries, play a crucial role (Roberts et al, 2008). In addition, they serve as important resources for other studies, such as assessing level of ambiguities of a language, and disambiguation (Maynard et al, 2004).

Because building and maintaining high quality gazetteers by hand is very time consuming (Kazama and Torisawa, 2008), many solutions have proposed generating gazetteers automatically from existing resources. In particular, the success that solutions which exploit Wikipedia[1] have been enjoying in many other NLP applications has encouraged a number of research works on automatic gazetteer generation to use Wikipedia, such as works by Toral and Muñoz (2006), and Kazama and Torisawa (2007).

Unfortunately, current systems still present several limitations. First, none have exploited the full content and structure of Wikipedia articles, but instead, only make use of the article's first sentence. However, the full content and structure of Wikipedia carry rich information that has been proven useful in many other NLP problems, such as document classification (Gabrilovich and Markovitch, 2006), entity disambiguation (Bunescu and Paşca, 2006), and semantic relatedness (Strube and Ponzetto, 2006). Second, no other works have evaluated their methods in the context of entity extraction tasks. Evaluating these generated gazetteers in real NLP applications is important, because the quality of these gazetteers has a major impact on the performance of NLP applications that make use of them. Third, the majority of approaches focus on newswire domain and the four classic entity types location (LOC), person (PER), organization (ORG) and miscellaneous (MISC), which have been studied extensively. However, it has been argued that entity extraction is often much harder in scientific domains due to complexity of domain languages, density of information and specificity of classes (Murphy et al, 2006; Byrne, 2007; Nobata et al, 2000).

In this paper we propose a novel approach to automatically generating gazetteers using external knowledge resources. Our method is language- and domain- independent, and scalable. We show that the content and various structural elements of Wikipedia can be successfully exploited to generate high quality gazetteers. To assess gazetteer quality, we evaluate it in the context of entity extraction in the scientific domain of Archaeology, and demonstrate that the generated gazetteers improve the performance of an SVM-based entity tagger across all entity types on an archaeological corpus.

The rest of the paper is structured as follows. In the next section, we review related work. In section 3 we explain our methodology for auto-

[1] http://en.wikipedia.org

Proceedings of the 2009 Workshop on the People's Web Meets NLP, ACL-IJCNLP 2009, pages 1–9,
Suntec, Singapore, 7 August 2009. ©2009 ACL and AFNLP

matic gazetteer generation. Section 4 introduces the problem domain and describes the experiments conducted. Section 5 presents and discusses the results. Finally we conclude with an outline of future work.

2 Related Work

Currently, existing methods to automatic gazetteer generation can be categorized into two mainstreams; *pattern driven approach* and *knowledge resource approach.*

The *pattern driven approach* uses domain- and language specific patterns to extract candidate entities from unlabeled corpora. The idea is to include features derived from unlabeled data to improve a supervised learning model. For example, Riloff and Jones (1999) introduced a bootstrapping algorithm which starts from seed lists and, iteratively learns and refines domain specific extraction patterns for a semantic category that are then used for building dictionaries from unlabeled data. Talukdar et al (2006), also starting with seed entity lists, apply pattern induction to an unlabeled corpus and then use the induced patterns to extract candidate entities from the corpus to build extended gazetteers. They showed that using the token membership feature with the extended gazetteer improved the performance of a Conditional Random Field (CRF) entity tagger; Kozareva (2006) designed language specific extraction patterns and validation rules to build Spanish location (LOC), person (PER) and organization (ORG) gazetteers from unlabeled data, and used these to improve a supervised entity tagger.

However, the *pattern driven approach* has been criticized for weak domain adaptability and inadequate extensibility due to the specificity of derived patterns. (Toral and Muñoz, 2006; Kazama and Torisawa, 2008). Also, often it is difficult and time-consuming to develop domain- and language-specific patterns.

The *knowledge resource approach*, attempts to solve these problems by relying on the abundant information and domain-independent structures in existing large-scale knowledge resources. Magnini et al (2002) used WordNet as a gazetteer together with rules to extract entities such as LOC, PER and ORG. They used two relations in WordNet; Word_Class, referring to concepts bringing external evidence; and Word_Instance, referring to particular instances of those concepts. Concepts belonging to Word_Class are used to identify trigger words for candidate entities in corpus, while concepts of Word_Instance are used directly as lookup dictionaries. They achieved good results on a newswire corpus. The main limitation of WordNet is lack of domain specific vocabulary, which is critical to domain specific applications (Schütze and Pedersen, 1997). Roberts et al (2008) used terminology extracted from UMLS as gazetteers and tested it in an entity extraction task over a medical corpus. Contrary to WordNet, UMLS is an example of a domain specific knowledge resource, thus its application is also limited.

Recently, the exponential growth in information content in Wikipedia has made this Web resource increasingly popular for solving a wide range of NLP problems and across different domains.

Concerning automatic gazetteer generation, Toral and Muñoz (2006) tried to build gazetteers for LOC, PER, and ORG by extracting all noun phrases from the first sentences of Wikipedia articles. Next they map the noun phrases to WorldNet synsets, and follow the hyperonymy hierarchy until they reach a synset belonging to the entity class of interest. However, they did not evaluate the generated gazetteers in the context of entity extraction. Due to lack of domain specific knowledge in WordNet, their method is limited if applied to domain specific gazetteer generation. In contrast, our method overcomes this limitation since it doesn't rely on any resources other than Wikipedia. Another fundamental difference is that our method exploits more complex structures of Wikipedia.

Kazama and Torisawa (2007) argued that while traditional gazetteers map word sequences to predefined entity categories such as "London → {LOCATION}", a gazetteer is useful as long as it returns consistent labels even if these are not predefined categories. Following this hypothesis, they mapped Wikipedia article titles to their hypernyms by extracting the first noun phrase after *be* in the first sentence of the article, and used these as gazetteers in an entity extraction task. In their experiment, they mapped over 39,000 search candidates to approximately 1,200 hypernyms; and using these hypernyms as category labels in an entity extraction task showed an improvement in system performance. Later, Kazama and Torisawa (2008) did the same in another experiment on a Japanese corpus and achieved consistent results. Although novel, their method in fact bypasses the real problem of generating gazetteers of specific entity types. Our method is essentially different in this aspect. In addition, they only use the first sentence of Wikipedia articles.

3 Automatic Gazetteer Generation – the Methodology

In this section, we describe our methodology for automatic gazetteer generation using the *knowledge resource approach*.

3.1 Wikipedia as the knowledge resource

To demonstrate the validity of our approach, we have selected the English Wikipedia as the external knowledge resource. Wikipedia is a free multilingual and collaborative online encyclopedia that is growing rapidly and offers good quality of information (Giles, 2005). Articles in Wikipedia are identified by unique names, and refer to specific entities. Wikipedia articles have many useful structures for knowledge extraction; for example, articles are inter-connected by hyperlinks carrying relations (Gabrilovich and Markovitch, 2006); articles about similar topics are categorized under the same labels, or grouped in lists; categories are organized as taxonomies, and each category is associated with one or more parent categories (Bunescu and Paşca, 2006). These relations are useful for identifying related articles and thus entities, which is important for automatic gazetteer generation. Compared to other knowledge resources such as WordNet and UMLS, Wikipedia covers significantly larger amounts of information across different domains, therefore, it is more suitable for building domain-specific gazetteers. For example, as of February 2009, there are only 147,287 unique words in WordNet[2], whereas the English Wikipedia is significantly larger with over 2.5 million articles. A study by Holloway (2007) identified that by 2005 there were already 78,977 unique categories divided into 1,069 disconnected category clusters, which can be considered as the same number of different domains.

3.2 The methodology

We propose an automatic gazetteer generation method using Wikipedia article contents, hyperlinks, and category structures, which can generate entity gazetteers of any type. Our method takes input seed entities of any type, and extends them to more complete lists of the same type. It is based on three hypotheses;

1. Wikipedia contains articles about domain specific seed entities.
2. Using articles about the seed entities, we can extract fine-grained *type labels* for them, which can be considered as a list of hypernyms of the seed entities, and predefined entity type hyponyms of the seeds.
3. Following the links on Wikipedia articles, we can reach a large collection of articles that are related to the source articles. If a related article's *type label* (as extracted above) matches any of those extracted for seed entities, we consider it a similar entity of the predefined type.

Naturally, we divide our methods into three steps; firstly we match a seed entity to a Wikipedia article (the matching phase); next we label seed entities using the articles extracted for them and build a pool of fine-grained *type labels* for the seed entities (the labeling phase); finally we extract similar entities by following links in articles of seed entities (the expansion phase). The pseudo-algorithm is illustrated in Figure 1.

3.2.1 Matching seed entities to Wikipedia article

For a given seed entity, we firstly use the exact phrase to retrieve Wikipedia articles. If not found, we use the leftmost longest match, as done by Kazama and Torisawa (2007). In Wikipedia, searches for ambiguous phrases are redirected to a Disambiguation Page, from which users have to manually select a sense. We filter out any matches that are directed to disambiguation pages. This filtering strategy is also applied to step 3 in extracting candidate entities.

3.2.2 Labeling seed entities

After retrieving Wikipedia articles for all seed entities, we extract fine-grained *type labels* from these articles. We identified two types of information from Wikipedia that can extract potentially reliable labels.

[2] According to http://wordnet.princeton.edu/man/wnstats.7WN , February 2009

```
Input: seed entities SE of type T
Output: new entities NE of type T
STEP 1 (section 3.2.1)
  1.1. Initialize Set P as articles for SE;
  1.2. For each entity e: SE
  1.3.     Retrieve Wikipedia article p for e;
  1.4.     Add p to P;
STEP 2 (section 3.2.2)
  2.1. Initialize Set L
  2.2. For each p: P
  2.3.     Extract fine grained type labels l;
  2.4.     Add l to L;
STEP 3 (section 3.2.3)
  3.1. Initialize Set HL;
  3.2. For each p: P
  3.3.     Add hyperlinks from p to HL;
  3.4. If necessary, recursively crawl extracted
         hyperlinks and repeat 3.2 and 3.3
  3.5. For each link hl: HL
  3.6.     Extract fine grained type labels l';
  3.7.     If L contains l'
  3.8.         Add title of hl to NE;
  3.9.         Add titles of redirect links of hl to
```

Figure 1. The proposed pseudo-algorithm for gazetteer generation from the content and various structural elements of Wikipedia

As Kazama and Torisawa (2007) observed, in the first sentence of an article, the head noun of the noun phrase just after *be* is most likely the hypernym of the entity of interest, and thus a good category label. There are two pitfalls to this approach. First, the head noun may be too generic to represent a domain-specific label. For example, following their approach the label extracted for the archaeological term “Classical Stage”[3] from the sentence “The Classic Stage is an *archaeological term* describing a particular developmental level.” is “term”, which is the head noun of “archaeological term”. Clearly in such case the phrase is more domain-specific. For this reason we use the exact noun phrase as category label in our work. Second, their method ignores a correlative conjunction which in most cases indicates equivalently useful labels. For example, the two noun phrases in *italic* in the sentence “Sheffield is a *city* and *metropolitan borough* in South Yorkshire, England” are equally useful labels for the article “Sheffield”. Therefore, we also extract the noun phrase connected by a correlative conjunction as the label. We apply this method to articles retrieved in 3.2.1. For simplicity, we refer to this approach to labeling seed entities as ***FirstSentenceLabeling***, and the labels created as L_s. Note that our method is essentially different from Kazama and Torisawa as we do not add these extracted nouns to gazetteers; instead, we only use them for guiding the extraction of candidate entities, as described in section 3.2.3.

As mentioned in section 3.1, similar articles in Wikipedia are manually grouped under the same categories by their authors, and categories are further organized as a taxonomy. As a result, we extract category labels of articles as fine-grained *type labels* and consider them to be hypernyms of the entity’s article. We refer to this method as ***CategoryLabeling,*** and apply it to the seed entities to create a list of category labels, which we denote by L_c.

Three situations arise in which the ***CategoryLabeling*** introduces noisy labels. First, some articles are categorized under a category with the same title as the article itself. For example, the article about “Bronze Age” is categorized under category “Bronze Age”. In this case, we explore the next higher level of the category tree, i.e., we extract categories of the category “Bronze Age”, including “2nd Millennium”, “3rd millennium BC”, “Bronze”, “Periods and stages in Archaeology”, and “Prehistory”. Second, some categories are meaningless and for management purposes, such as “Articles to be Merged since 2008”, “Wikipedia Templates”. For these, we manually create a small list of “stop” categories to be discarded. Third, according to Strube and Ponzetto (2008), the category hierarchy is sometimes noisy. To reduce noisy labels, we only keep labels that are extracted for at least 2 seed entities.

Once a pool of fine-grained *type labels* have been created, in the next step we consider them as fine-grained and immediate hypernyms of the seed entities, and use them as *control vocabulary* to guide the extraction of candidate entities.

3.2.3 Extracting candidate entities

To extract candidate entities, we first identify from Wikipedia the entities that are related to the seed entities. Then we select from them those candidates that share one or more common hypernyms with the seed entities. The intuition is that in the taxonomy, nodes that share common immediate parents are mostly related, and, therefore, good candidates for extended gazetteers.

[3] Any Wikipedia examples for illustration in this paper make use of the English Wikipedia, February 2009, unless otherwise stated.

We extract related entities by following the hyperlinks from the articles retrieved for the seed entities, as by section 3.2.1. This is because in Wikipedia, articles often contain mentions of entities that also have a corresponding article, and these mentions are represented as outgoing hyperlinks. They link the main article of an entity (*source entity*) to other sets of entities (*related entities*). Therefore, by following these links we can reach a large set of related entities to the seed list. To reduce noise, we also filter out links to disambiguation pages as in section 3.2.1. Next, for each candidate in the related set, we use the two labeling approaches introduced in section 3.2.2 to extract its *type labels*. If any of these are included by the *control vocabulary* built with the same labeling approach, we accept them into the extended gazetteers. That is, if the *control vocabulary* is built by ***FirstSentenceLabeling*** we only use ***FirstSentenceLabeling*** to label the candidate. The same applies to ***CategoryLabeling***. One can easily extend this stage by recursively crawling the hyperlinks contained in the retrieved pages. In addition, some Wikipedia articles have one or more redirecting links, which groups several surface forms of a single entity. For example a search for “army base” is redirected to article “military base”. These surface forms can be considered as synonyms, and we thus also select them for extend gazetteers.

After applying the above processes to all seed entity articles, we obtain the output extended gazetteers of domain-specific types. To eliminate potentially ambiguous entities, for each extended gazetteer, we exclude entities that are found in domain-independent gazetteers. For example, we use a generic person name gazetteer to exclude ambiguous person names from the extended gazetteers for LOC.

4 Experiments

In this section we describe our experiments. Our goal is to build extended gazetteers using the methods proposed in section 3, and test them in an entity extraction task to improve a baseline system. First we introduce the setting, an entity extraction task in the archaeological domain; next we describe data preparation including training data annotation and gazetteer generation; then, we introduce our baseline; and finally present the results.

4.1 The Problem Domain

The problem of entity extraction has been studied extensively across different domains, particularly in newswire articles (Talukdar et al 2006), bio-medical science (Roberts et al, 2008). In this experiment, we present the problem within the domain of archaeology, which is a discipline that has a long history of active fieldwork and a significant amount of legacy data dating back to the nineteenth century and earlier. Jeffrey et al (2009) reports that despite the existing fast-growing large corpora, little has been done to develop high quality meta-data for efficient access to information in these datasets, which has become a pressing issue in archaeology. To our best knowledge, three works have piloted the research on using information extraction techniques for automatic meta-data generation in this field. Greengrass et al (2008) applied entity and relation extraction to historical court records to extract names, locations and trial names and their relations; Amrani et al (2008) used a series of text-mining technologies to extract archaeological knowledge from specialized texts, one of these tasks concerns entity extraction. Byrne (2007) applied entity and relation extraction to a corpus of archaeology site records. Her work concentrated on nested entity recognition of 11 entity types.

Our work deals with archaeological entity extraction from un-structured legacy data, which mostly consist of full-length archaeological reports varying from 5 to over a hundred pages. According to Jeffrey et al (2009), three types of entities are most useful to an archaeologist;

- Subject (SUB) – topics that reports refer to, such as findings of artifacts and monuments. It is the most ambiguous type because it covers various specialized domains such as warfare, architecture, agriculture, machinery, and education. For example “Roman pottery”, “spearhead”, and “courtyard”.
- Temporal terms (TEM) – archaeological dates of interest, which are written in a number of ways, such as years “1066 - 1211”, “circa 800AD”; centuries “C11”, “the 1st century”; concepts “Bronze Age”, “Medieval”; and acronyms such as “BA” (Bronze Age), “MED” (Medieval).
- Location (LOC) – place names of interest, such as place names and site addresses related to a finding or excavation. In our study, these refer to UK-specific places.

Source	Domain	Tag Density
astro-ph	Astronomy	5.4%
MUC7	Newswire	11.8%
GENIA	Biomedical	33.8%
AHDS-selected	Archaeology	9.2%

Table 1. Comparison of tag density in four test corpora for entity extraction tasks. The "AHDS-selected" corpus used in this work has a tag density comparable to that of MUC7

4.2 Corpus and resources

We developed and tested our system on 30 full length UK archaeological reports archived by the Arts and Humanities Data Service (AHDS)[4]. These articles vary from 5 to 120 pages, with a total of 225,475 words. The corpus is tagged by three archaeologists, and is used for building and testing the entity extraction system. Compared to other test data reported in Murphy et al (2006), our task can be considered hard, due to the heterogeneity of information of the entity types and lower tag density in the corpus (the percentage of words tagged as entities), see Table 1. Also, according to Vlachos (2007), full length articles are harder than abstracts, which are found common in biomedical domain. This corpus is then split into five equal parts for a five-fold cross validation experiment.

For seed gazetteers, we used the MIDAS Period list[5] as the gazetteer for TEM, the Thesaurus of Monuments Types (TMT2008) from English Heritage[6] and the Thesaurus of Archaeology Objects from the STAR project[7] as gazetteers for SUB, and the UK Government list of administrative areas as the gazetteer for LOC. In the following sections, we will refer to these gazetteers as *GAZ_original*.

4.3 Automatic gazetteer generation

We used the seed gazetteers together with the methods presented in section 3 to build new gazetteers for each entity type, and merge them with the seeds as extended gazetteers to be tested in our experiments. Since we introduced two methods for labeling seed entities (section 3.2.2), which are also used separately for selecting extracted candidate entities (section 3.2.3), we design four experiments to test the methods separately as well as in combination; specifically for each entity type, *GAZ_EXT*$_{firstsent}$ denotes the extended gazetteer built using ***FirstSentenceLabeling*** for labeling seed entities and selecting candidate entities; *GAZ_EXT*$_{category}$ refers to the extended gazetteer built with ***CategoryLabeling***; *GAZ_EXT*$_{union}$ merges entities in two extended gazetteers into a single gazetteer; while *GAZ_EXT*$_{intersect}$ is the intersection of *GAZ_EXT*$_{firstsent}$ and *GAZ_EXT*$_{category}$ i.e., taking only entities that appear in both. Table 2 lists statistics of the gazetteers and Table 3 displays example type labels extracted by the two methods.

To implement the entity extraction system, we used Runes[8] data representation framework, a collection of information extraction modules from T-rex[9], and the machine learning framework Aleph[10]. The core of the tagger system is a SVM classifier. We used the Java Wikipedia Library[11] (JWPL v0.452b) and the Wikipedia dump of Feb 2007 published with it.

4.4 Feature selection and baseline system

We trained our baseline system by tuning feature sets used and the size of the token window to consider for feature generation; and we select the best performing setting as the baseline. Later we add official gazetteers in section 4.1 and extended gazetteers as in section 4.3 to the baselines and use gazetteer membership as an additional feature to empirically verify the improvement in system accuracy.

The baseline setting thus used a window size of 5 and the following feature set:

- Morphological root of a token
- Exact token string
- Orthographic type (e.g., lowercase, uppercase)
- Token kind (e.g., number, word)

4.5 Result

Table 4 displays the results obtained under each setting, using the standard metrics of *Recall (R)*, *Precision (P)* and *F-measure* (*F1)*. The bottom row illustrates **I**nter **A**nnotator **A**greement *(IAA)*

[4] http://ahds.ac.uk/

[5] http://www.midas-heritage.info and http://www.fish-forum.info

[6] http://thesaurus.english-heritage.org.uk

[7] http://hypermedia.research.glam.ac.uk/kos/STAR/

[8] http://runes.sourceforge.net/

[9] http://t-rex.sourceforge.net/

[10] http://aleph-ml.sourceforge.net/

[11] http://www.ukp.tu-darmstadt.de/software/jwpl/

	LOC	**SUB**	**TEM**
GAZ_original	11,786 (8,228 found)	5,725 (4,320 found)	61 (43 found)
GAZ_EXT $_{\textit{\textbf{firstsent}}}$	19,385 (7,599)	11,182 (5,457)	163 (102)
GAZ_EXT $_{\textit{\textbf{category}}}$	18,861 (7,075)	13,480 (7,745)	305 (245)
GAZ_EXT $_{\textit{\textbf{union}}}$	23,741 (11,955)	16,697 (10,972)	333 (272)
GAZ_EXT $_{\textit{\textbf{intersect}}}$	14,022 (2,236)	7,455 (1,730)	133 (72)

Table 2. Number of unique entities in each gazetteer, including official and extended versions. GAZ_EXT includes GAZ_original. For GAZ_original, numbers in brackets are the number of entities found in Wikipedia. For others, they are the number of extracted entities that are new to the corresponding GAZ_original

LOC		**SUB**		**TEM**	
FirstSentence-Labeling (597)	*CategoryLabeling* (779)	*FirstSentence-Labeling* (1342)	*CategoryLabeling* (761)	*FirstSentence-Labeling* (11)	*CategoryLabeling* (10)
village, small village, place, town, civil parish	villages in north Yorkshire, north Yorkshire geography stubs, villages in Norfolk, villages in Somerset, English market towns	facility, building, ship, tool, device, establishment	ship types, monument types, gardening, fortification, architecture stubs	period, archaeological period, era, century, millennium	Periods and stages in archaeology, Bronze age, middle ages, historical eras, centuries

Table 3. Top 5 most frequently extracted (counted by number of seed entities sharing that label) fine-grained *type labels* for each entity type. Numbers in brackets are the number of unique labels extracted

	LOC			**SUB**			**TEM**		
	P	R	F1	P	R	F1	P	R	F1
Baseline (B)	*69.4*	*67.4*	*68.4*	*69.6*	*62.3*	*65.7*	*82.3*	*81.4*	*81.8*
B+ GAZ_original	*69.0*	*72.1*	*70.5*	*69.7*	*65.4*	*67.5*	*82.3*	*82.7*	*82.5*
B+ GAZ_EXT $_{\text{firstsent}}$	**69.9**	**76.7**	**73.1**	**70.0**	**68.3**	**69.1**	**82.6**	**84.6**	**83.6**
B+ EXT $_{\text{category}}$	**69.1**	**75.1**	**72.0**	68.8	**67.0**	**67.9**	82.0	**83.7**	**82.8**
B+ EXT $_{\text{union}}$	68.9	**75.0**	**71.8**	**69.8**	**66.5**	**68.1**	**82.4**	**83.4**	**82.9**
B+ EXT $_{\text{intersect}}$	69.3	**76.2**	**72.6**	**69.7**	**67.6**	**68.6**	**82.6**	**84.3**	**83.4**
IAA	-	-	75.3	-	-	63.6	-	-	79.9

Table 4. Experimental results showing accuracy of systems in the entity extraction task for each type of entities, varying the feature set used. Baseline performances are marked in *italic*. Better performances than baselines achieved by our systems are highlighted in *bold*.

between the annotators on a shared sample corpus of the same kind as that for building the system, calculated using the metric by Hripcsak and Rothschild (2005). The metric is equivalent to scoring one annotator against the other using the *F1* metric, and in practice system performance can be slightly higher than *IAA* (Roberts et al, 2008). The *IAA* figures for all types of entities are low, indicating that the entity extraction task for the archaeological domain is difficult, which is consistent with Byrne (2007)'s finding.

5 Discussion

As shown in Table 2, our methods have generated domain specific gazetteers that almost doubled the original seed gazetteers in every occasion, even for the smallest seed gazetteer of TEM. This proves our hypotheses formulated in section 3.1, that by utilizing the hyperonymy relation and exploring information in an external resource, one can extend a gazetteer by entities of similar types without utilizing language- and domain-specific knowledge. Also by taking the intersection of entities generated by the two labeling methods (bottom row of table 2), we see that the overlap is relatively small (from 30%-40% of the list generated by either method), indicating that the extended gazetteers produced by the two methods are quite different, and may be used to complement each other. Combining figures in Table 3, we see that both methods extract fine-grained type-labels that on average extract 4 - 14 candidate entities.

The quality of the gazetteers can be checked using the figures in Table 4. First, all extended gazetteers improved over the baselines for the three entity types, with the highest increase in *F1* of 4.7%, 3.4% and 1.8% for LOC, SUB, and

TEM respectively. In addition, they all outperform the original gazetteers, indicating that the quality of extended gazetteers is good for the entity extraction task.

By comparing the effects of each extended gazetteer, we notice that using the gazetteers built with type-labels extracted from the first sentence of Wikipedia article always outperforms using those built via the Wikipedia categories, indicating that the first method (***FirstSentenceLabeling***) results in better quality gazetteers. This is due to two reasons. First, the category tree in Wikipedia is not a strict taxonomy, and does not always contain *is-a* relationships (Strube and Ponzetto, 2006). Although we have eliminated categories that are extracted for only one seed entity, the results indicate the extended gazetteers are still noisier than those built by ***FirstSentenceLabeling***. To illustrate, the articles for SUB seed entities "quiver" and "arrowhead" are both categorized under "Archery", which permits noisy candidates such as "Bowhunting", "Camel archer" and "archer". Applying a stricter filtering threshold may resolve this problem. Second, compared to Wikipedia categories, the labels extracted from the first sentences are sometimes very fine-grained and restrictive. For example, the labels extracted for "Buckinghamshire" from the first sentence are "ceremonial Home County" and "Non-metropolitan County", both of which are UK-specific LOC concepts. These rather restrictive labels help control the gazetteer expansion within the domain of interest. The better performance with ***FirstSentenceLabeling*** indicates that such restrictions have played a positive role in reducing noise in the labels generated, and then improving the quality of candidate entities.

We also tested effects of combining the two approaches, and noticed that taking the intersection of gazetteers generated by the two approaches outperform the union, but figures are still lower than the single best method. This is understandable because by permitting members of noisier gazetteers the system performance degrades.

6 Conclusion

We have presented a novel language- and domain- independent approach for automatically generating domain-specific gazetteers for entity recognition tasks using Wikipedia. Unlike previous approaches, our approach makes use of richer content and structural elements of Wikipedia. By applying this approach to a corpus of the Archaeology domain, we empirically observed a significant improvement in system accuracy when compared with the baseline systems, and the baselines plus original gazetteers.

The extensibility and domain adaptability of our methods still need further investigation. In particular, our methods can be extended to introduce several statistical filtering thresholds to control the label generation and candidate entity extraction in an attempt to reduce noise; also the effect of recursively crawling Wikipedia articles in the candidate extraction stage is worth studying. Additionally, it would be interesting to study other structures of Wikipedia, such as list structures and info boxes, in gazetteer generation. In future we will investigate into these possibilities, and also test our approach in different domains.

Acknowledgement

This work is funded by the Archaeotools[12] project that is carried out by Archaeology Data Service, University of York, UK and the Organisation, Information and Knowledge Group (OAK) of University of Sheffield, UK.

References

Ahmed Amrani, Vichken Abajian, Yves Kodratoff, and Oriane Matte-Tailliez. 2008. A Chain of Text-mining to Extract Information in Archaeology. In *Proceedings of Information and Communication Technologies: From Theory to Applications, ICT-TA 2008*, 1-5.

Razva Bunescu and Marius Paşca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of EACL2006*

Kate Byrne. Nested Named Entity Recognition in Historical Archive Text. In *Proceedings of International Conference on Semantic Computing*, 2007.

Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 1301-1306, Boston, 2006.

Jim Giles. Internet Encyclopedias Go Head to Head. In *Nature 438*. 2005. 900-901.

Mark Greengras, Sam Chapman, Jamie McLaughlin, Ravish Bhagdev and Fabio Ciravegna. Finding Needles in Haystacks: Data-mining in Distributed Historical Datasets. In *The Virtual Representation of the Past*. London, Ashgate. 2008

[12] http://ads.ahds.ac.uk/project/archaeotools/

George Hripcsak and Adam S. Rothschild. Agreement, the F-measure and Reliability in Information Retrieval: In *Journal of the American Medical Informatics Association*, 296-298. 2005

Todd Holloway, Miran Bozicevic and Katy Börner. Analyzing and Visualizing the Semantic Coverage of Wikipedia and its Authors. In *Complexity, Volumn 12, issue 3*, 30-40. 2007

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. To appear in *special Theme Issues of the Philosophical Transactions of the Royal Society A,"Crossing Boundaries: Computational Science, E-Science and Global E-Infrastructures"*.

Jun'ichi Kazama and Kentaro Torisawa. 2008. Inducing Gazetteers for Named Entity Recognition by Large-scale Clustering of Dependency Relations. In *Proceedings of ACL-2008: HLT*, 407-415.

Jun'ichi Kazama and Kentaro Torisawa. Exploting Wikipedia as External Knowledge for Named Entity Recognition. In *Proceedings of EMNLP-2007 and Computational Natural Language Learning 2007*. 698-707.

Zornista Kozareva. 2006. Bootstrapping Named Entity Recognition with Automatically Generated Gazetteer Lists. *In EACL-2006-SRW.*

Bernardo Magnini, Matto Negri, Roberto Prevete and Hristo Tanev. AWordNet-Based Approach to Named Entity Recognition. In *Proceedings of COLING-2002 on SEMANET: building and using semantic networks*. 1-7

Diana Maynard, Kalina Bontcheva and Hamish Cunningham. Automatic Language-Independent Induction of Gazetteer Lists. In *Proceedings of LREC2004*.

Tara Murphy, Tara Mcintosh and James R Curran. Named Entity Recognition for Astronomy Literature. In *Proceedings of the Australasian Language Technology Workshop*, 2006.

Chikashi Nobata, Nigel Collier and Jun'ichi Tsujii. Comparison between Tagged Corpora for the Named Entity Task. In Proceedings of the Workshop on Comparing Corpora at ACL2000.

Ellen Riloff and Rosie Jones. 1999. Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 474-479.

Angus Roberts, Robert Gaizauskas, Mark Hepple and Yikun Guo. Combining Terminology Resources and Statistical Methods for Entity Recognition: an Evaluation. In *Proceedings of LREC2008*.

Hinrich Schütze and Jan O. Pedersen. A cooccurrence-based thesaurus and two applications to Information Retrieval. In *Information Processing and Management: an International Journal, 1997*. 33(3): 307-318

Michael Strube and Simone Paolo Ponzetto. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006. 1419 - 1424

Partha Pratim Talukdar, Thorsten Brants, Mark Liberman and Fernando Pereira. 2006. A Context Pattern Induction Method for Named Entity Extraction. In *Proceedings of CoNLL-2006,* 141-148.

Antonio Toral and Rafael Muñoz. 2006. A Proposal to Automatically Build and Maintain Gazetteers for Named Entity Recognition by using Wikipedia. In *Proceedings of Workshop on New Text, 11th Conference of the European Chapter of the Association for Computational Linguistics 2006*.

Andreas Vlachos. Evaluating and Combining Biomedical Named Entity Recognition Systems. In *Workshop: Biological translational and clinical language processing*. 2007