PolyUHK: A Robust Information Extraction System for Web Personal Names

Ying Chen
The Hong Kong Polytechnic University
+852-3400-3272
chenying3176@gmail.com

Sophia Yat Mei Lee
The Hong Kong Polytechnic University
+852-3400-3926
sophiaym@gmail.com

Chu-Ren Huang
The Hong Kong Polytechnic University
+852-2766-4832
churenhuang@gmail.com

ABSTRACT
Personal information extraction is an important component of advanced information retrieval. There are two problems needed to be solved in this practical task: personal name ambiguity and extraction of personal information for a specific person. For personal name ambiguity, which is a very common phenomenon in the fast growing Web resource, we propose a robust system which extracts features with a totally unsupervised approach from resources beyond the given Web corpus. The experiments show that these broad features not only can improve performances, but also increase the robustness of a disambiguation system. For personal information extraction, a rule-based information extraction system is introduced, which is able to re-use current well-developed tools effectively and identify the properties of Web data. The experiments show that the system can achieve state-of-the-art performances, especially the high precision.

Categories and Subject Descriptors
H.3.5 [INFORMATION STORAGE AND RETRIEVAL]: Online Information Systems, *Web-based Service;* H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing*;* I.2.7 [Artificial Intelligence]: Natural Language Processing

General Terms
Algorithms, Performance, Experimentation

Keywords
Information Extraction, Attribute Extraction, Disambiguation, Person Names, Web Documents

1. INTRODUCTION
Extraction of Information for a specific named entity is of growing importance due to the needs of an increasing number of commercial applications ranging from the presentation of basic search results to the automatic compilation of a specific named entity. In general, information extraction (IE) consists of two main subtasks: the detection of the interested named entity and the extraction of information for the specific named entity. Since named-entity ambiguity is very common in Web data, the detection of the given named entity becomes a big challenge. Moreover, due to the noisy unstructured text data in the Web, information extraction is also difficult.

Web People Search (WePS) (Artiles et al., 2009; Sekine and Artiles, 2009) provides a forum for a standard evaluation, which focuses on IE for personal named-entities in Web data. It includes two tasks: clustering and attribute extraction (AE). The clustering task, which can also be called personal name disambiguation, groups those web pages according to whether the given personal name appearing in that webpage refers to the same person in reality. Attribute extraction, which can be considered as a special case of IE, extracts certain personal information for a focus person with the given personal name

For personal name disambiguation, many of the systems, which participated in the WePS 2007 bakeoff, use a combination of feature extraction followed by clustering to disambiguate names. The features employed include simple tokens, base syntactic chunks, named entities, dependency parses, semantic role labels, etc. For the most part, these features are extracted using off-the-shelf components designed to annotate or extract the relevant information. Unfortunately, Web data are quite diverse and differ fundamentally from the news-oriented sources that have traditionally been the source of training material for many of the NLP systems used in feature extraction. This leads to a severe degradation in the performance of the individual feature extractors and the subsequent clustering algorithms. In this paper, we explore an approach designed to overcome these difficulties: lightweight features from web-derived resources.

Attribute extraction for a specific named entity is still a difficult task even in one document. In general, attribute extraction has two key components: named-entity recognition (NER) and relation detection and recognition (RDR). Since most of the current NER and RDR systems are developed only for news articles (formal style), their adoption to Web data (informal style) is not an easy task (Vilain et.al, 2007). Similar to the genre categories – "formal text" and "informal text" – defined in Minkov et al. (2005), text formats are categorized as "formal style" and "informal style". "Formal style" follows restrictive writing standards and usually uses complete sentences, while "informal style" has few limitations on writing format and can mix various representations. To effectively handle the combination of formal and informal style text in Web data, this paper presents a rule-based AE system, which can also incorporate some relation patterns specific for Web data.

The remainder of this paper is organized as follows. Section 2 addresses the related work of both personal name disambiguation and AE. Section 3 introduces our personal name disambiguation approach and the rule-based AE system. Section 4 presents the experiments and analysis of our disambiguation system and AE system for the WePS 2009 corpus. Finally, some conclusions are drawn.

2. Previous work
2.1 Clustering
Due to the varying ambiguity of personal names in a corpus, most previous disambiguation systems use unsupervised clustering as the basic approach, despite the different features used to create the

similarity space. There are two kinds of features: document-level features and global features. Document-level features refer to the information extracted from the given corpus, such as tokens in a given webpage (Bagga and Baldwin, 1998; Gooi and Allan, 2004; Pedersen et al., 2005), biographical information (Mann, 2006), and bigrams (Pedersen & Kulkarni, 2007). Global features refer to information derived beyond the given corpus, such as information from an extra corpus (Mann, 2006; Kalmar and Blume, 2007; Pedersen and Kulkarni; 2007), online information (Rao et al., 2007), and so on. Global feature extraction is comparatively less popular than document-level feature extraction.

Mann (2006) and Kalmar and Blume (2007) find that the given corpus is often not large enough to learn the realistic probabilities or weights for those features, and therefore they added more data into the given corpus in order to give a realistic probability for a token (Mann, 2006) or a realistic frequency for a proper name (Kalmar & Blume, 2007). However, Pedersen & Kulkarni (2007) use an extra corpus to filter out some un-collocation bigrams in the given corpus, and then create a token-based representation for the bigram feature with the help of scores that measure whether a bigram is a collocation.

Besides the given corpus, there is a large amount of online information about the focus person. Since most of the given corpora include only a small part of documents containing the ambiguous personal names, such as about the top 100 webpages in the WePS 2009 corpus, it is sometimes hard to make a co-reference decision, even for an annotator. To alleviate this shortcoming, for each ambiguous personal name, Rao et al. (2007) retrieve one thousand snippets from the query of the personal name in the Google search engine, and merge these low-noise snippets into the training or test data. Each snippet is treated as a document, and can be served as a bridge. For example, if two webpages referring to the same named entity have not enough shared information, the additional snippet-based document can sometimes provide a bridge to connect them.

2.2 Attribute extraction

It is well-known that Web corpus is heterogeneous and noisy, such as texts with mixed usage of both formal and informal style, noisy capitalization information and so on. Therefore, most of the current NER and RDR systems can not work well because they heavily rely on surface cues, which are usually noisy in Web data. Here, we develop a novel algorithm that segments a webpage into fragments according to their writing style: formal style and informal style, and then use a rule-based system, in which a pattern is limited to a fragment.

In addition, web data also have different ways of conveying certain information. For example, it is common that occupation and affiliation information is expressed in a homepage in the format of "Name, Position, Affiliation," such as "Anita Coleman, Assistant Professor, University of Arizona." As this kind of web-specific expression is often multi-lines, some existing IE patterns (Rosenfeld and Feldman, 2006; Mann and Yarowsky, 2003; Mann 2006), which are limited to one sentence or are designed for formal style text, can not directly be applied. To catch this Web expression property, we develop some patterns working on fragments, and the experiment shows that those patterns could achieve high precision, which is very important for real applications.

WWW 2009, April 20-24, 2009, Madrid, Spain.

3. Methodology

3.1 The clustering system

First, we need to define the object to be disambiguated. Our system makes the standard "one person per document" assumption where all mentions of a target ambiguous personal name in a document are assumed to refer to the same entity in reality, and therefore that all the features associated with each mention of that name can be combined. In the WePS 2009 Web corpus, this assumption usually holds with the exception of genealogy pages.

Then, for each page, we employ some simple preprocessing as a precursor to feature extraction: Beautiful Soup[1] (a HTML parser) is run to extract a clean text document for that webpage, and then each clean document is run through MXTERMINATOR[2] to find sentence boundaries. Finally, features are extracted which include token-based features, n-gram features, and snippet features..

3.1.1 Token-based features

Simple token-based features are used in almost every disambiguation system. There are various places to extract the tokens of interest, such as the body of a given webpage, the title of a given webpage and so on. Here, we extract four kinds of tokens: Queryname tokens, Full tokens, URL tokens, and Title tokens in root page (TTRP).

> ***Queryname tokens***: the tokens occurring in sentences that include a mention of the ambiguous personal name;
> ***Full tokens***: the tokens occurring in a given webpage;
>
> ***URL tokens***: the tokens occurring in the corresponding URL of a given webpage;
>
> ***Title tokens in root page (TTRP):*** Tokens occurring in the title of the root page of a given webpage.

Due to the somewhat noisy information in URL tokens and TTRP tokens, we combine them with Full tokens: for each URL token and TTRP token, if the token is also one of the Full tokens of other webpages, this token is added into the Full token list of the current webpage.

Three token-based features are formed from these base sets of tokens – Queryname tokens, Full tokens, and Full* tokens (containing Full tokens, URL tokens, and TTRP tokens). Then, each token in each feature vector is weighed by using a TFIDF weighting scheme defined as follows.

Given a feature vector S, $S = (s_1, \ldots, s_n)$, s_i $(i = 1, \ldots, n)$ are substrings (tokens). For a substring (token) w in S, its TFIDF weight is computed as:

$$V(w, S) = \frac{V'(w, S)}{\sqrt{\sum_{w \in S} V'(w, S)^2}}$$

$$V(w, S) = \log(TF_{w,S} + 1) \times \log(IDF_w)$$

[1] http://www.crummy.com/software/BeautifulSoup

[2] http://www.id.cbs.dk/~dh/corpus/tools/MXTERMINATOR

where $TF_{w,S}$ is the frequency of substring w in S, and IDF_w is the inverse of the fraction of webpages in the given corpus that contain w.

3.1.2 N-gram features

A disambiguation system that merely uses the extracted tokens often cannot fully solve the problem of Web personal name ambiguity. One reason for the poor performance of tokens alone for a Web corpus is noisy Web tokens, which are derived mainly from the free writing style text in a webpage. For example, a token, "professor," in the Web 1T 5-gram corpus, which was released by Google, can occur in different shapes: Professor.Cozort, Professor97, and so on. If the given corpus is too small, it is impossible for the above TFIDF weighting scheme to learn the realistic frequencies of "professor" or to learn the realistic weight of this token, which is, however, very important for disambiguation. Since it is hard to avoid noisy tokens in a Web corpus, we choose to add a large extra Web corpus that can reflect a realistic weighting distribution among Web tokens and re-learn the weights for Web tokens of interest.

Although the extraction of syntactic and semantic phrases in previous work often cannot work well for a Web corpus, a phrase, a unit larger than a word, often includes a large amount of information useful for disambiguation, such as a proper name. Due to the lack of high-quality tools that can extract phrases for a webpage, an unsupervised method, which is similar to the one used in Pedersen & Kulkarni (2007), is used to extract bigrams in a webpage.

The large Web corpus used here is the Web 1T 5-gram corpus, which was collected by Google in January 2006 from approximately 1 trillion words of text from publicly accessible English webpages. This corpus contains all n-grams occurring in those webpages and their corresponding frequencies. The length of the n-grams ranges from one to five grams.

To use the Web 1T 5-gram corpus, necessary preprocessing needs to be done for a given webpage. For each sentence, the Google tokenization is used to get the tokens in that sentence. In addition, stop tokens in the given webpage are marked with a stop-word dictionary in order to reduce search time in the Web 1T 5-gram corpus. The Google tokenization is very similar to the Penn Treebank tokenization.

Unigram feature: Since the weight for a token in Full tokens often is unrealistic due to the small size of the given corpus and noisy tokens in a Web corpus, we reweight using the Web 1T 5-gram corpus.

To learn the real weight for a token (a non-stop token) w, first the frequency of this token is gotten by searching w in the Web 1T 5-gram corpus, and then its weight is calculated by the above TFIDF formula, except that $V'(w, S)$ is redefined as follows:

$$V(w, S) = \log(TF_{w,S} + 1) \times (\log(N/TF_{w,gc}) + \log(IDF_w))$$

Where $TF_{w,S}$ is the frequency of substring w in S (the webpage where w is located); $TF_{w,gc}$ is the frequency of substring w in the Web 1T 5-gram corpus; N is the sum of the frequencies of all unigrams occurring in the Web 1T 5-gram corpus; and IDF_w is the inverse of the fraction of webpages in the given corpus that contain w. The above formula still follows the main idea of a TFIDF weighting scheme: the higher probability ($TF_{w,gc}$/N) a token has in the Web 1T 5-gram corpus, the less important (weighty) the token is for disambiguation.

Bigram feature: Although a bigram often does not include as much information as a proper name does, its extraction is unsupervised and does not need any labeled data which is easy to be applied in a Web corpus.

For each sentence in a webpage, all possible bigrams are extracted, except the bigrams that include a stop token or punctuation. The removal of those bigrams not only can filter out noisy phrases, but also can effectively save the search time in the Web 1T 5-gram corpus. Then, the frequencies of the remaining bigrams are determined by searching them in the Web 1T 5-gram corpus.

Even though some noisy bigrams is removed, some remaining bigrams may have no meaning at all and therefore add a noise to disambiguation. Therefore, only the bigrams that are true collocations are kept in the disambiguation system. There are several methods to test whether an bigram is a collocation in the study of Pedersen & Kulkarni (2007), namely Fisher's Exact Test, the Log-Likelihood Ratio, the Odds Ratio, and Pointwise Mutual Information. These methods perform similarly, hence our system chooses the Log-Likelihood ratio method with the binomial distribution (Manning and Schütze, 1999) to filter out un-collocation bigrams.

Even though the Web 1T 5-gram corpus is created from such a large Web corpus, the data sparsity problem still cannot be avoided. In the WePS corpus, there are many bigrams that cannot be found in the Web 1T 5-gram corpus, and thus a smoothing solution is needed. Most state-of-the-art smoothing schemes use context-dependent backoff, such as Kneser-Ney Smoothing (Kneser & Ney, 1995). To reduce the computation in the disambiguation system, an overly simple backoff is used for unknown bigrams in our system. In the Web 1T 5-gram corpus, a unigram appearing only 200 times or more is kept, and an n-gram (except a unigram) appearing only 40 times or more is kept. So the simple backoff works as follows: the frequency of an unknown unigram is 199 and the frequency of an unknown bigram is 39. The normalization problem is ignored for now.

Given the log-likelihood ratios of all reserved bigrams, an experimental threshold is chosen from the training data, and the bigrams whose log-likelihood ratio is greater than the threshold (this means that the bigram is not a collocation) are filtered out. Then, for each remaining bigram, its weight is learned with the modified TFIDF weighting scheme as was done for the unigram feature above. Notice that N is the sum of the frequencies of all bigrams occurring in the Web 1T 5-gram corpus.

3.1.3 Snippet-based features

An extra large Web corpus can ameliorate the problem of unrealistic TFIDF weighting learning for noisy Web tokens and the low-quality phrase extraction, but it cannot overcome the insufficient information problem in the given corpus. However,

besides the given dataset of an ambiguous personal name, there is a large amount of online information available that focuses only on the ambiguous personal name. The disambiguation system tries to use the snippet information from a search engine as in Rao et al. (2007), but with a different strategy.

After n-gram feature extraction, for each webpage, all useful bigrams have been extracted by using the Web 1T 5-gram corpus and the log-likelihood ratio method, and their corresponding weights have been learned by the modified TFIDF weighting scheme. Now, for each webpage, more information about its ambiguous object is retrieved by using those bigrams and a search engine as follows:

- For each webpage, select the bigram with the maximal weight, query to a search engine (Yahoo is chosen) with the string: the bigram + the ambiguous personal name, and keep the top 100 snippets provided in the returned webpage. The bigram with the maximal weight is often an important phrase in that webpage to describe the ambiguous object.
- Exclude those snippets that do not include the query string, and then concatenate all remaining snippets into a new document. This new document often contains some extra information about the ambiguous object, which may not be provided in the given webpage.
- Do a tokenization for the new document and create a token-based feature vector. Then, use the TFIDF weighting scheme that is used in the simple token-based features to learn the weight for each token.

The snippet-based feature extracts the token information from snippets directly available from the search retrieval, which often includes more information about the ambiguous object beyond the given webpage. Certainly, there is more information in the webpages from the search retrieval, but it is efficient to use only the snippets because downloading those webpages takes time.

3.2 The attribute extraction system

Our AE framework consists of two main components: preprocessing (webpage type detection and fragment segmentation) and personal information extraction.

3.2.1 Preprocessing

Given a webpage, it is first categorized into three webpage types according to its relationship with the focus person: homepage, related webpage (webpage mainly describes the focus person, such as biographical webpage), and others. It is then segmented into several fragments based on its text styles, which could be formal style fragment or informal style fragment. Figure 1 gives examples of these two fragments expressing the similar information.

Formal style fragrance

Anita Sundaram Coleman is an Assistant Professor in the School of Information Resources & Library Science at the University of Arizona, Tucson, which she joined in 2001.

Informal style fragrance

Anita Coleman

Assistant Professor

School of Information Resources & Library Science

1515 E. First St.

University of Arizona

Tucson, AZ 85719

Figure 1: Examples of two kinds of fragrances

```
if    title of a webpage contains a keyword for "homepage"
            web type =  "homepage"
elif  title is the person name:
            web type =  "homepage"
elif title contains the person name
            web type =  "related page"
elif the URL of a webpage contains the last name or the
                first name of the personal name:
            web type = "homepage"
else:
            web type = "others"
```

Figure 2: The algorithm of webpage type detection

3.2.1.1 Webpage type detection

Webpage type often provides some information for AE. For example, "I" in a homepage refers to the focus person, and all information in that sentence is about the focus person. In this study, we apply some naïve rules to a webpage and its URL to detect its webpage type. The detail is presented in Figure 2.

3.2.1.2 Text fragment segmentation

It is common that a webpage is often written in a mixture of different representations: formal style and informal style. For example, in a resume, the description of "objective" is usually in formal style, while the "education experience" section is more likely to be in informal style. There are two steps in fragment segmentation. First, each line is classified into two classes – formal style or informal style – according to the percentage of tokens beginning with capitalization. Second, continuous lines that share the same expression type are linked into a fragment.

Pattern 1: personal name, occupation (linkword affiliation)?

Pattern 2: personal name
occupation (linkword affiliation)?

Pattern 3: personal name
occupation
affiliation

* "linkword" is a comma or a preposition * "(linkword affiliation)?" means affiliation appears at most once

Figure 3. Web-specific patterns

Table 1. List of the attributes in WePS 2009 data

Attribute names					
date of birth	birth place	other name	occupation	affiliation	relatives
phone	fax	email	website	nationality	
degree	major	school	mentor	award	

3.2.2 Personal Information Extraction

As mentioned, AE includes two components: NER and RDR. Our current AE system focuses on a rule-based NER and RDR. To detect named entity expressions, a rule-based NER, which usually has low performance, is used for all attributes. Given the named-entity expressions detected by a NER system, their relationships to the focus person are detected by a rule-based RDR system. As explained, the unit, which our NER and RDR systems work on, is a fragment. Take the advantage of the information of webpage type, we apply our NER and RDR systems to all fragments if the webpage is a homepage, and use the NER and RDR only to the fragments that contains the focus personal name if the webpage is not a homepage.

Because informal style text is often noisy, currently we use only patterned and keywords to extract named entity expressions that we are interested in. For example, we collect "occupation" keywords from "Dictionary of Occupational Titles" (DOT) and use organization keywords from GATE.

Given the named-entity expressions detected by the rule-based NER system, their relationships to the focus person are detected using the following rule: whether a keyword appears in that sentence. Each attribute has its own set of keywords. For example, keyword "born" is chosen for the attributes of "date of birth" and "birth place."

We also create some rule-based patterns for the whole fragment so as to identify web-specific expressions. Figure 3 shows how the occupation and affiliation attributes are extracted from the informal style example in Figure 1.

4. Experiments

We run all experiments on WePS 2009 corpus. For clustering, there are 79 personal names in the training data and 30 personal names in the test data. For attribute extraction, there are 18 personal names in the training data and 30 personal names in the test data. The data set for each personal name consists of about 100 webpages that contain the focus personal name.

The WePS 2009 corpus provides a real corpus which can test a disambiguation system for personal names with different frequency (varying ambiguity) and in different domains. It can also fairly evaluate the robustness of a disambiguation system, which is very important for a real application. Personal names in this corpus are selected from multiple domains, such as the US census, English Wikipedia, the Program Committee listing of a Computer Science conference (ECDL-2006), and ACL-2006 conference, so that it can reflect the real data distribution in different fields.

For AE task, 16 kinds of attributes (Table 1) could be extracted from a webpage if existing. We also find that the personal attributes varies from the contact information, such as "phone," "fax," and "email," to some complicated information, such as "award" and "relatives." These various attributes often require different information extraction technologies. For example, contact information often can be detected by simple patterns, "affiliation" and "occupation," however, needs more advanced AE methods.

4.1 Experiments of the clustering system

Since the parameter setting for the clustering system is very important, we focus only on the purity-based scoring (Artiles et al., 2009), and acquire an overall optimal fixed stop-threshold from the training data, and then use it in test data. In this section, we report our results evaluated by the clustering scoring provided by WePS 2009 evaluation, which includes both the B-cubed scoring (Bagga & Baldwin, 1998) and the purity-based scoring. In this experiment, for each feature model, the results of the disambiguation system are evaluated, and the average performances (F scores) are listed in Table 2 for the test data of the WePS 2009 corpus. The feature model in the experiment begins only with Queryname tokens, and then continues to add the five features, Full tokens, Full* tokens, the unigram feature, the bigram feature, and the snippet-based feature from a search engine, one by one. Moreover, Table 2 shows the F scores calculated with different weights (0.2 and 0.5) to precision/recall or P/IP.

Table 2. Performances of our clustering systems on the WePS 2009 test data

	BEP-BER (0.5)	BEP-BER (0.2)	P-IP (0.5)	P-IP (0.2)
Queryname tokens	45	82	62	84
+ Full tokens	***90***	***83***	49	80
+ Full* tokens	89	***83***	48	80
+ unigram	84	79	***89***	86
+ bigram	87	87	87	81
+ snippet	82	80	88	***87***

Table 3. Performances for each attributes of our AE system on the WePS 2009 test data

("same" means the performance is same as above)

	affiliation	occupation	birth place	date of birth	other name	degree
Rule-based AE	5.9	12.4	15.2	11.8	37.4	30.5
	email	fax	phone	website	award	
Rule-based AE	43.1	48.5	33.2	16.0	13.9	

Table 2 shows that the performance with fixed stop-thresholds is improved by incorporating more features. A big improvement is achieved by adding the unigram feature, which uses the Web 1T 5-gram corpus to learn the realistic weights. This means that although the given corpus can provide most information for disambiguation, it still is impossible for a disambiguation system to understand and utilize all this document-level information without the help of broader biographical knowledge learned from other broad resources. At the same time, we can also notice that the performances with fixed stop-thresholds have become stable after adding the unigram feature, even the personal names in those data have varying ambiguity. This indicates that our broad feature extraction is robust enough for different personal names.

Meanwhile, because the clustering result is often sensitive to parameter settings, it is not surprising to find that some performances have inconsistent changes for some feature models. For example, the incorporation of the Full token feature drops the P-IP (0.5) performance by 13% (forth column in Table 2), but increases 45% for the BEP-BER (0.5) score (second column in Table 2). Therefore, for a real application, it is necessary to find a robust feature model for all names. In Table 2, we can notice that the final feature model "Queryname tokens + Full* tokens + unigram + bigram + snippet" is comparably robust no matter which scoring algorithm is used, B-cubed scoring or purity-based scoring.

In general, noisy features generated by feature extractors and insufficient information hurt the robustness of a disambiguation system. Results show that n-gram features effectively ameliorate the low-quality phrase extraction, which often causes the problem of insufficient information or noisy information for disambiguation. In addition, they also lessen the bad effects from unrealistic weighting learning, which, again, adds noisy information for disambiguation. The URL-related information in the Full* token feature and the snippet-based feature can provide more information about the ambiguous object so that they can partially solve the insufficient information problem for disambiguation.

4.2 Experiments of the AE system

We run our rule-based AE systems for the test data of the WePS 2009 corpus. Our AE system is the best AE system in the WePS 2009 evaluation, and achieves the best F score: 14.11. The low F score also indicates that personal information extraction from Web data is a very difficult task. Therefore, more effort on this is needed. Comparing to the precision and recall (36.22 vs. 8.76), we also notice that the rule-based AE system has achieved very high precision, which is promising for real applications.

Table 3 shows the detail performances (F scores) of the 11 kinds of attributes that we mainly focus on. It is not surprising that the attributes of "email," "fax," and "phone" achieve very good performances because they are almost fixed expressions, whereas the attributes of "affiliation," "occupation," and "award" do not perform well because our rules can not catch their various ways of expression. Hence, more work should be done in this regard.

5. Conclusion

In this paper, we report a robust personal disambiguation system and a rule-base attribute extraction system designed for a Web corpus. Both systems are developed towards real applications of IE.

Our personal disambiguation system extracts features with various lightweight methods and from broad resources, such as downloading related Web pages, using an large Web corpus (the Web 1T 5-gram corpus), and extracting the information from a search engine. Although these unsupervised broad features do not capture much attention until now, our experiment shows that those features can achieve a comparable or even better performance than the ones extracted with more complex supervised NLP tools.

We also explore the robustness of a disambiguation system, which is very important for real applications. It is very difficult for a disambiguation system to work consistently well for all personal names with varying ambiguity. Our experiments show that the collection of the unsupervised features extracted from broad resources can effectively improve the robustness of a disambiguation system, but more work is needed on robust

clustering that can learn an optimal clustering setting for each personal name.

Our rule-based AE system achieves start-of-the-art performance for Web personal information extraction. However, the total performance is still very low, which indicates the problem of Web personal information extraction is far from being solved and more work is needed.

From the analysis of the AE performances, we find that it is very important to have a high-quality NER system for both formal and informal style text. However, this is still a big challenge, specifically for informal style text in Web data because of noisy information. Meanwhile, we will try to incorporate some existing NER and RDR system into our system in the future. Finally, the question as to how to effectively extract and use web-specific information in personal information extraction, such as webpage type and web-specific patterns, also needs more exploration.

6. REFERENCES

[1] Artiles, Javier, Julio Gonzalo and Satoshi Sekine, "The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task," In Proceedings of Semeval 2007, Association for Computational Linguistics, 2007.

[2] Artiles, Javier, Julio Gonzalo and Satoshi Sekine. "WePS 2 Evaluation Campaign: overview of the Web People Search Clustering Task," In 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009.

[3] Bagga, Amit and Breck Baldwin, "Entity-based Cross-document Co-referencing Using the Vector Space Model," In Proceedings of the 17th International Conference on Computational Linguistics, 1998.

[4] Chen, Ying and James H. Martin, "CU-COMSEM: Exploring Rich Features for Unsupervised Web Personal Name Disambiguation," In Proceedings of Semeval 2007, Association for Computational Linguistics, 2007.

[4] Chen, Ying, James Martin and Martha Palmer, "Robust Disambiguation of Web-based Personal Names," In Proc. of Second IEEE International Conference on Semantic Computing, 2008.

[5] Culotta, A., R. Bekkerman, and A. McCallum, "Extracting social networks and contact information from email and the web," CEAS-04, 2004.

[6] Elmacioglu, Ergin, Yee Fan Tan, Su Yan, Min-Yen Kan and Dongwon Lee, "PSNUS: Web People Name Disambiguation by Simple Clustering with Rich Features," In Proceedings of Semeval 2007, Association for Computational Linguistics, 2007.

[7] Etzioni, O., M. Cafarella, et al., "Unsupervised named-entity extraction from the Web: An experimental study," Artificial Intelligence, 2005.

[8] Gooi, Chung Heong and James Allan, "Cross-Document Coreference on a Large Scale Corpus," In Proceedings of NAACL-2004, 2004.

[9] Kalmar, Paul and Matthias Blume, "FICO: Web Person Disambiguation Via Weighted Similarity of Entity Contexts," In Proceedings of Semeval 2007, Association for Computational Linguistics, 2007.

[10] Kneser, Reinhard and Hermann Ney, "Improved backing-off for m-gram language modeling," In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pages 181- 184, 1995.

[11] Li, Xin, Paul Morie, and Dan Roth, "Robust Reading: Identification and Tracing of Ambiguous Names," In Proceedings of NAACL, pp. 17—24, 2004.

[12] Mann, G. and D. Yarowsky, "Unsupervised Personal Name Disambiguation," CoNLL, 2003.

[13] Mann, Gideon S., "Multi-Document Statistical Fact Extraction and Fusion," PhD Thesis, 2006.

[14] Manning, Christopher and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA., 1999.

[15] Minkov, E., R. Wang, and W. Cohen. "Extracting Personal Names from Emails: Applying Named Entity Recognition to Informal Text," HLT/EMNLP, 2005.

[16] Niu, Cheng, Wei Li, and Rohini K. Srihari, "Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction," In ACL, 2004.

[17] On, Byung-Won and Dongwon Lee, "Scalable Name Disambiguation using Multi-Level Graph Partition," SIAM, 2007.

[18] Pedersen, Ted, Amruta Purandare, and Anagha Kulkarni, "Name Discrimination by Clustering Similar Contexts," In Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico, 2005.

[19] Pedersen, Ted and Anagha Kulkarni, "Unsupervised Discrimination of Person Names in Web Contexts," In the Proceedings of the Eighth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, 2007.

[20] Popescu, Octavian and Bernardo Magnini, "IRST-BP: Web People Search Using Named Entities," In Proceedings of Semeval 2007, Association for Computational Linguistics, 2007.

[21] Rao, Delip, Nikesh Garera and David Yarowsky, "JHU1 : An Unsupervised Approach to Person Name Disambiguation using Web Snippets," In Proceedings of Semeval 2007, Association for Computational Linguistics, 2007.

[22] Rosenfeld, B. and R. Feldman, "URES : an Unsupervised Web Relation Extraction System," COLING/ACL, 2006.

[23] Rosenfeld, B., R. Feldman, et al., “TEG: a hybrid approach to information extraction,” CIKM, 2004.

[24] Sekine, Satoshi and Javier Artiles. “WePS 2 Evaluation Campaign: overview of the Web People Search Attribute Extraction Task,” In 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, 2009.

[25] Vilain, M., J. Su, and S. Lubar, “Entity Extraction is a Boring Solved Problem ¨C or is it?.,” Association for Computational Linguistics, 2007.