Person attribute extraction from the textual parts of Web pages

István T. Nagy[1], Richárd Farkas[2],

[1] Universitiy of Szeged, Department of Informatics
[2] Research Group of Artificial Intelligence, Hungarian Academy of Sciences
{nistvan, rfarkas}@inf.u-szeged.hu

Abstract. We present the RGAI systems which participated in the third Web People Search Task challenge. The chief characteristics of our approach are that we focus on the raw textual parts of the Web pages instead of the structured parts, we group similar attribute classes together and we explicitly handle their interdependencies. The RGAI systems achieved top results on the attribute extraction subtask, and average results on the clustering subtask.

Keywords: natural language processing, information extraction, Web content mining, person attribute extraction, document clustering

1 Introduction

Personal names are among the most frequently searched items in Web search engines. At the same time these types of search results ignore the fact that one name may be related to more than one person. Sometimes person names are highly ambiguous. The first WePS challenge organized in 2007 focused on this disambiguation problem. As input, the participants' systems received Web pages retrieved from a Web search engine using a given person name as a query. The aim of the task was to find all the different people among the results lists and assign a corresponding document to each person. During the evaluation of WePS1, the organizers realized that some attributes are very useful for the person disambiguation problem. Hence the second WePS challenge organized in 2009 contained an absolutely new challenge. The attribute extraction subtask was to identify 16 different attributes from Web pages such as birth date, affiliation, and occupation. This subtask proved very difficult and the best system only achieved an F-measure score of 12.2. The third WePS shared task introduced a novel subtask which sought to mine attributes for persons, i.e. the attribute extraction from the clusters of pages belonging to each given person. We will now describe our system that participated in this third WePS challenge.

2 Related work

The aim of Web Content Mining is to extract useful information from the natural language-written parts of websites. The first attempts on Web Content Mining began with the Internet began around 1998-'99 [3], [4], [5], [6]. They were expert systems with hand-crafted rules or indued rules used in a supervised manner and based on labeled corpora. The next generation of approaches on the other hand works in weakly-supervised settings [7],[8],[9]. Here the input is a seed list of target information pairs and the goal is to gather a set of pairs which are related to each other in the same manner as the seed pairs. The pairs may contain related entities (for example, country - capital city in [7] and celebrity partnerships in [10]) or form an entity-attribute pair (like Nobel Prize recipient - year in [11]) or may be concerned with retrieving all available attributes for entities [9]. These systems generally download Web pages that contain the seed pairs, then learn syntactic / semantic rules from the sentences of the pairs (they generally use the positive instances for one case as negative instances for another case).

The person name disambiguation subtask of the second WePS challenge [1] was dominated by systems which had a preprocessing step, where the HTML documents were converted to the plain texts then standard clustering algorithms were employed with a bag of words representation of the pages. The participants of the attribute extraction subtask of this challenge [2] generally used hand-crafted rules for the attribute classes. Named Entity Recogniser was also applied here but with combination of pre and postprocessing heuristics (the best performing system used only expert rules).

3 Our methods

We shall focus on the raw text parts of the Web pages because we found that more pages express content in textual form than in structured form [16]. The first step of information extraction may be to construct a good section selection module. When handling the problem, we first extract the candidate attributes from the relevant sections of Web pages, then we cluster the pages by merging clusters having common person attributes and aggregate attributes to the persons identified.

3.1 Preprocessing

The input of the participants' system was a set of pages retrieved from a Web search engine using a given person's name as a query. We assumed that useful information is available in the natural language-written part of websites and tables [16]. This is why we concentrated on the natural language-written part of websites and tables, and we discarded a lot of noisy and misleading elements from pages (e.g. menu elements). These elements can seriously hinder the proper functioning of Natural Language Processing (NLP) tools.

In order to identify textual paragraphs we applied the Stanford POS tagger for each section of the DOM tree of the HTML files. We assumed that one piece of text was a textual paragraph if it was longer than 60 characters and it contained more than one verb. We extracted several attributes with our own Named Entity Recognition (NER) [14] system which was trained on CoNLL-2003 training data sets. When we used this model on the entire set of paragraphs, the accuracy score obtained was low. To handle this problem we developed attribute-specific, relevant section selection modules. Firstly we looked for the occurrences of all gold standard attributes using simple string matching in each extracted paragraph. In this way we created a database with 'positive' and 'negative' paragraphs for the actual attribute. Then we created a set of positive words with the most frequently occurring words from the positive paragraphs. If a paragraph in the prediction phase contained at least one word from the actual positive list, we marked it as a positive paragraph and we only extracted attributes from these paragraphs. This approach was used to find the *occupation*, *affiliation*, *award* and *school* attributes.

3.2 Attribute extraction

Like other WePS2 systems [12],[13], our attribute extraction system also consists of two fundamental parts: the candidate attributes extraction module and an attribute verification module. Based on this approach, we first mark potential attribute values in a paragraph. Second, we find out which candidate values are exhibited.

When handling this subtask of attribute extraction, it seems necessary to classify the attribute classes in several ways. First, we aggregated similar attributes into logical groups. For instance, the name group contains the *other name*, *relatives* and *mentor* attribute classes On the other, we can assume subordinate relations among the coherent attributes. For example, we only marked a candidate name as *mentor* if it was not *relatives* or *other name*.

Table 1. Attribute typologies

Name	Availability	Organization
relatives	e-mail	school
other name	web page	award
mentor	phone number	affiliation
	fax	

Next, we will elaborate on the extraction procedure for each of the attributes.

Date of birth: if a paragraph contains *born*, *birth* or *birthday* phrases we find candidate dates with a date validator within a window of the word. This validator works with 9 different regular expressions rules, and can identify dates written in different formats in the span of text.

Birth place: when a paragraph contains *born*, *birth*, *birthplace*, *hometown* and *native* phrases we use the location markups given by the NER tool [14] trained on the locations class of the CoNLL-2003 training dataset to identify candidate locations for

the birthplace. We accept a location as a birthplace if a birthplace validator validates it.

Occupation: according to the WePS2 results, it was one of the most difficult, ambiguous and frequent attribute classes, which is due to the abstract nature of this attribute. Hence we avoided using lists. It is not available in any NER model or training database. So we created a training database by matching all gold annotation to paragraphs. We used simple string matching and we did not know where the actual attribute occurred. However, the resulting dataset was very noisy. We trained our NER tool [14] on this training database, and we used it on the candidate occupation paragraphs.

Organizations (school, award, affiliation): we found that these types of attributes were names of organizations so we grouped them together. We also used an NER tool [14] here trained on the organization class of the CoNLL-2003 training data to identify candidate organization mentions only in affiliation-candidate paragraphs. When the NER model marks a candidate organization phrase, we first search for the *school* attribute. Then a potential candidate organization is marked as a school if it appears near some cue phrases such as *graduate*, *degree*, *attend*, *education* and *science*. Next we defined a school validator that uses the MIVTU [12] school word frequency list with *School*, *High*, *Academy*, *Christian*, *HS*, *Central* and *Senior*. We extended this list with *University*, *College*, *Elementary*, *New*, *State*, *Saint*, *Institute* phrases. First letter capitalized sequences, except for some stopwords like *of* and *at* which contain at least one of these words were marked as a school by a validator. If the school validator did not validate the potential candidate organization, we looked for the award attribute. When candidate sequences appear near cue phrases such as *award*, *win*, *won*, *receive* and *price*, we assumed an expression with award was an attribute. We also defined an award validator, which validates a first letter capitalized sequence except for some stopword like *at* and *of*, if it contains at least one element of the *award*, *prize*, *medal*, *order*, *year*, *player* and *best* phrases. When the candidate string is not a valid school and award, we tag it to the affiliation attribute.

Degree: a list of degrees complied manually which contains 62 items. When we found one element from these lists in a paragraph we marked it as a degree attribute. We assumed that the degree attribute might be located far from the name in a CV-type Web page.

Names (*relatives*, *other name*, *mentor*): these types of attributes are person names so we found that they occur together. To identify name attributes we used an NER tool [14] trained on the person names of the CoNLL training data. A model extracts name phrases as *relatives* if they appear in the immediate context of the candidate that indicates various relationships like *father*, *son*, *daughter* and so on. Cue phrases were the same as in the MIVTU [12] system used in WePS2 and are also available in Wikipedia. Sometimes we did not mark the potential candidate sequence for *relatives*, but looked for *other name* attributes instead. We hypothesized that a person does not write his or her name using the same number of tokens; at the same time *other name* has to contain at least a part of the original name. This hypothesis may not be true for nicknames. For example when the original name was *Helen Thomas*, we did not accept the candidate string *Helen McCumber*, but we accepted the *Helen M. Thomas* sequence. If a name was not marked as relatives or other name, we analyzed the potential candidates for mentor name. If it appeared near cue phrases such as *study*

with, work with, coach, train, advis, mentor, supervisor, principal, manager and *promote* we marked the potential candidate sequence as a mentor attribute.
Nationality: We created a list of nationalities that contained 371 elements. It has multiple entries for certain nationalities. Once we found one element from this list in a paragraph or table, we assumed a potential nationality attribute. Then we selected the most frequent potential nationality attribute of the Web pages.

When extracting *availability* attribute classes we did not used just the textual paragraphs, but examined the whole text of Web pages as these types of attributes may occur in other parts as well.
Phone: when a text contains *tel, telephone, ph:, phone, mobile, call, reached at, office, cell* or *contact* words or a part of the original name, we applied the following regular expression:

```
(((?[0-9+(][.()0-9s/-]{4,}[0-9])((?(s?x|s?ext|s?hart).?)? d{1,5})?)
```

It is a permitted regular expression for potential phone numbers. We defined a phone number validator that validated the sequence determined by the regular expression.
Fax: we use the same method as for phone numbers, i.e. we look for *fax, telfax* and *telefax* phrases.
E-mail: we assumed that if somebody offers their e-mail address, it is also a link. Therefore, we examined links that contain the mailto tag. Moreover, we assumed that every mail address contains the original name or one part of the original name. Hence we defined an e-mail address validator that validates e–mail addresses. We generate all character trigrams from the original name and when an e-mail address contains at least one of them, the validator accepts it. We defined a stop list as well. This list contains words such as *wiki, support*, and *webmaster*. Should a candidate e-mail address contain one from the stop list, the validator does not accept it. Next we extracted the domain from all accepted e-mail addresses, which we used for the web-site attribute.
Web-site: we assumed that when somebody displays a Web address on a website, it is also a link too, so a Web address is a link at the same time. In this case we only extract a website attribute from links. We marked a potential candidate attribute as a website when it contained the original name or one part of the original or extracted domain name from the e-mail attribute.

3.3 Clustering

Our chief hypothesis in the clustering subtask was that it can be effectively solved by using extracted person attributes. We defined a weighting of attribute classes. The most useful attribute classes were w*eb address*, *e-mail, telephone, fax number* and *other name* and they got a weight of 3. In addition, we weighted *birth date* as 2 while *birth place*, *mentor*, *affiliation*, *occupation*, *nationality*, *relatives*, *school*, and *award* each got a weight of 1. Then every document was represented by a vector with extracted person attribute values.

To define a document similarity measure, we needed to normalize the attribute values, i.e. spelling variants and synonyms have to be handled as equivalents. We developed individual normalization rules for each attribute class. For example, the birth place of *United States of America* could be referred to as *USA*, *U.S.A. United States*, *Federal United States* and so on. Here, we created a synonym dictionary based on the re-direct links of the English Wikipedia and we developed regular expressions or transformation rules for other attribute classes.

As a first approach for Web page clustering, a bottom-up heuristic clustering was performed. Here the starting clusters consist of the individual Web pages and then the clusters are merged iteratively until a stopping criterion is reached. For each step of this procedure the most similar clusters are merged (the union of their attributes formed the attribute set of the resulting cluster), where the similarity measure of the weighted size of the intersection of the cluster attribute sets was employed using normalization rules. The stopping criterion was defined to be a similarity value threshold of 2, i.e. if the similarity value of the closest clusters is less than 2 the procedure is terminated (RGAI5 submission).

Besides this heuristic bottom-up clustering, we employed the Xmeans algorithm in the WEKA Java package [15] as well. The advantage of this approach is that it is not necessary to define the number of clusters, but we can define the minimum number of clusters. We used the final number of clusters obtained by the heuristic clustering as the minimum number of clusters for Xmeans. (RGAI3)

In addition to person attribute-based Web page clustering, we also experimented with a text-based approach. With the results of RGA1, we only used the search engine snippet data. These types of representation compress the most important pieces of information. We represented the dataset with the tf-idf vector space model where every document is a vector. The RGAI1 and the RGAI2 results were almost identical. Lastly, RGAI4 is a hybrid method of the above two approaches, i.e. the feature sets of the person-based attribute and the snippet-based clustering were merged.

3.4 Attribute aggregation

As a last step, we had to aggregate those attributes that occurred in Web pages and were found in a cluster of pages, i.e. belonged to a particular person. The official evaluation metric of the challenge required only one attribute from each class. As we extracted more than one potential attribute values for each class, we had to choose one (e.g. a person may mention several of his affiliations). In the end we chose the most frequent element per person from each attribute class. When some attribute frequency was equal, we just chose it at random.

4 Results and discussion

Because the WePS3 attribute extraction subtask required clustering, the documents we submitted ran the attribute extraction and clustering tasks as well. The test dataset was composed of 300 person names and nearly 200 Web documents for each name.

The attributes had to be assigned to each person cluster rather than to individual pages. The training dataset was the WePS2 train and test sets, which contains 5,122 websites with 187,032 textual paragraphs. We found 2,781 affiliation, 3,419 occupation and 2,092 biographical paragraphs. For the location, organization and names, markups given by the NER tool [14] trained on the CoNLL-2003 training dataset achieved F_scores of 89.94 on names, 87.06 on locations and 76.37 on organizations.

During the evaluation of the clustering subtask the organizers used the extended versions of BCubed Precision and Recall, which was the official evaluation metric with alpha set to 0.5. They evaluated the clustering of documents for each query just focusing on two different people, except for 50 names, where only documents about one person were considered. The official results on the clustering task of the RGAI systems, the best performing participant and two baselines are shown in Table 1. Here our RGAI 1 system achieved the best scores.

Table 1. Document clustering results

System	avg. Bcubed precision	avg. Bcubed recall	avg. F-measure
YHBJ_2_unofficial	0.61	0.60	0.55
RGAI_AE_1	**0.38**	**0.61**	**0.40**
RGAI_AE_2	0.38	0.61	0.40
RGAI_AE_5	0.40	0.57	0.40
RGAI_AE_3	0.47	0.43	0.38
RGAI_AE_4	0.36	0.55	0.38
one_in_one_baseline	1.00	0.23	0.35
All_in_one_baseline	0.22	1.00	0.32

For the attribute extraction subtask[1], the evaluation metrics were computed as follows,
Precision: for a given person, it is the number of correct attribute/value pairs divided by the total number of attribute/value pairs extracted.
Recall: for a given person, it is the number of attributes having at least one correct value divided by the total number of attributes for which a correct value has been found by at least one of the systems.
F-measure: 1 / (alpha * 1/prec + (1-alpha) * 1/rec), where alpha was 0.5.
The above defined "given person" is taken from the prediction of the clustering subtask. The gold standard annotation of clustering consists of two person (clusters) for every document set. During the evaluation process the most similar predicted clusters was taken into account where the F score or recall was used as similarity metric, where
Precision: the number of documents in the cluster that refer to the person / number of documents in cluster.

1 Please note that at the time of preparing the workshop proceedings, official results for the attribute extraction subtask were not available due to unexpected difficulties of the task organizers with the manual assessments. Due the organizers, the results of the paper are achived on the 12.5 percent of the test dataset.

Recall: the number of documents in the cluster that refer to the person / number of documents that refer to the person.
Next, the organizers defined two different interpretations of the manual annotations, which were combined with the other two clustering evaluation options.
Strict evaluation: we count as correct all attribute / value pairs judged as correct by a majority of annotators and as incorrect otherwise.
Lenient evaluation: we count as correct all attribute / value pairs judged as correct or inexact by a majority of annotators, and as incorrect otherwise.
Table 2 shows the results of the RGAI systems when the clustering resemblance was the recall approach and the manual annotation was lenient. Our best result was achieved by the RGAI 3 system, but the Intelius system was outstanding.

Table 2 Lenient annotation and recall based clustering

System	Precision	Recall	F-measure
Intelius_AE_UNOFFICIAL	13.52	31.46	15.67
RGAI_AE_3	**5.02**	**5.01**	**4.38**
RGAI_AE_1	2.99	5.18	3.40
RGAI_AE_4	2.55	5.45	3.06
RGAI_AE_5	2.84	4.17	2.90
RGAI_AE_2	2.59	3.64	2.41

However, when we used the lenient annotation interpretation and the clustering approach based on the F score, our RGAI 3 system achieved significantly better results (see Table 3).

Table 3 Lenient annotation with F-measure based clustering

System	Precision	Recall	F-measure
RGAI_AE_3	**12.36**	**13.41**	**11.47**
Intelius_AE_UNOFFICIAL	9.13	15.07	9.77
RGAI_AE_2	6.91	7.93	6.32
RGAI_AE_1	6.64	7.53	6.08
RGAI_AE_4	4.85	7.56	5.31
RGAI_AE_5	5.31	6.07	4.76

When we used the strict annotation and recall-based clustering, the results of Intelius system were dramatically better than those of other systems. It was able to cluster the documents better (see Table 4**.**).

Table 4 Strict anotation with recall based clustering

System	Precision	Recall	F-measure
Intelius_AE_UNOFFICIAL	13.11	31.13	15.33
RGAI_AE_3	**4.75**	**4.88**	**4.25**
RGAI_AE_4	2.51	5.33	3.00
RGAI_AE_5	2.84	4.17	2.90
RGAI_AE_1	2.38	4.70	2.84

RGAI_AE_2	2.59	3.64	2.41

Finally, Table 5 shows the results when the clustering approach was based on the F score and the annotation was strict. RGAI 3 could achieved the best result systems performed fairly well.

Table 5 Strict annotation with F-measure based clustering

System	Precision	Recall	F-measure
RGAI_AE_3	**11,88**	**12,68**	**10,90**
Intelius_AE_UNOFFICIAL	9.33	14.94	9.73
RGAI_AE_2	7.06	8.24	6.53
RGAI_AE_1	5.99	7.79	5.65
RGAI_AE_4	4.78	7.99	5.36
RGAI_AE_5	5.06	5.88	4.54

The above tables show that our approach achieved an F score sligthly above 10 of F score based clustering. Compared to the WePS2 results – where the best system achieved about an of F score of twelve – these results are competitive as we solved a more complex problem here. On the other hand, the recall-based results show that our clustering approach has to be improved.

5 Summary and Conclusions

In this article we presented a person name disambiguation method with biographical attribute extraction from documents related to a person. We handled the name disambiguation problem from person Web search results. Our method is based on extracted biographical attributes and snippet information. The proposed clustering method was evaluated using the test dataset created for the name disambiguated subtask of the third Web People Search Task. Our clustering approach got an F score of 40 and was ranked fourth among the eight participants.

For the second subtask of the shared task, our method efficiently extracted the different types of attributes from Web pages and we achieved top results on the WePS3 challenge. We think that the reasons for the success of our attribute extractor are the followings. First, our approach groups attribute classes and introduces rules which efficiently handle the interdependencies among these classes. Second, we focused on the textual parts of the web pages using NLP tools which demonstrates that raw text parts of person Web pages should be analyzed besides the structured parts of the pages.

Acknowledgments

This work was supported in part by NKTH grant of the Jedlik Ányos R&D Programme (project codename TEXTREND) of Hungarian government. The authors would like to thank the shared task organizers for their devoted efforts.

6 References

1. Javier Artiles, 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, April 20th-24th, 2009, Madrid, Spain.
2. Satoshi Sekine 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, April 20th-24th, 2009, Madrid, Spain.
3. Brad Adelberg. 1998. Nodose - a tool for semiautomatically extracting structured and semistructured data from text documents. ACM SIGMOD, 27(2):283–294.
4. Mary Elaine Califf and Raymond J. Mooney. 1999. Relational learning of pattern-match rules for information extraction. In Proceedings of the Sixteenth National Conference on Artificial Intelligence, pages 328–334.
5. Dayne Freitag. 1998. Information extraction from html: Application of a general machine learning approach. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, pages 517–523.
6. Raymond Kosala and Hendrik Blockeel. 2000. Web mining research: A survey. SIGKDD Explorations, 2:1–15.
7. Oren Etzioni, Michael Cafarella, Doug Downey, Ana maria Popescu, Tal Shaked, Stephen Soderl, Daniel S. Weld, and Er Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. Artificial Intelligence, 165:91–134.
8. Satoshi Sekine. 2006. On-demand information extraction. In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 731–738, Sydney, Australia, July. Association for Computational Linguistics.
9. Kedar Bellare, Partha Talukdar, Giridhar Kumaran, Fernando Pereira, Mark Liberman, Andrew McCallum, and Mark Dredze. 2007. Lightly-supervised attribute extraction for web search. In Proceedings of NIPS 2007 Workshop on Machine Learning for Web Search.
10. Xiwen Cheng, Peter Adolphs, Feiyu Xu, Hans Uszkoreit, and Hong Li. 2009. Gossip galore – a selflearning agent for exchanging pop trivia. In Proceedings of the Demonstrations Session at EACL 2009, pages 13–16, Athens, Greece, April. Association for Computational Linguistics.
11. Hong Li Feiyu Xu, Hans Uszkoreit. 2007. A seeddriven bottom-up machine learning framework for extracting relations of various complexity. In Proceedings of ACL 2007, 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic
12. Keigo Watanabe and Danushka Bollegala. MIVTU: A Two-Step Approach to Extracting Attributes for People on the Web. In: 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, April 20th-24th, 2009, Madrid, Spain.
13. Xianpei Han and Jun Zhao. CASIANED: People Attribute Extraction based on Information Extraction. In: 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference, April 20th-24th, 2009, Madrid, Spain.
14. György Szarvas, Richárd Farkas, and András Kocsor. 2006. A multilingual named entity recognition system using boosting and C4.5 decision tree learning algorithms. DS2006, LNAI, 4265:267–278.

15.Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
17.István Nagy, Richárd Farkas and Márk Jelasity. Researcher affiliation extraction from homepages. NLPIR4DL ACL Workshop 2009 pp 1-9