UC3M at WePS2-AE: Acquiring Patterns for People Attribute Extraction from Webpages

César de Pablo-Sánchez
Universidad Carlos III de Madrid
Avda. de la Universidad Carlos III, 22
Colmenarejo, Spain
cdepablo@inf.uc3m.es

Paloma Martínez
Universidad Carlos III de Madrid
Avda. de la Universidad 30
Leganés, Spain
pmf@inf.uc3m.es

ABSTRACT

In this paper we describe the UC3M system for person attribute extraction that took part in the WePS2-AE task. In our system we applied Named Entity Recognition and Classification (NERC) to select candidate attributes from cleaned web pages. Then we applied patterns to select Named Entities that are correct attribute values. We have bootstrapped the acquisition of patterns from the training set of webpages. We also used the training extracted attribute as seeds and experimented with different alternatives for finding patterns. The task has proven to be harder than expected. Our best run performed a manual check on patterns before deploying them to the attribute extraction pipeline.

Categories and Subject Descriptors

H.3.1 [**Information Systems**]: Content Analysis and Indexing; H.3.3 [**Information Systems**]: Information Search and Retrieval; H.3.4 [**Information Systems**]: Systems and Software—*Question-Answering (fact retrieval) systems , World Wide Web (WWW)*

General Terms

Keywords

Information Extraction, Attribute Extraction, People Search, Semantic Search

1. INTRODUCTION

We present our experiments in the Web People Search Attribute Extraction (WePS-AE) task at the second WePS challenge. Briefly, the task requires to locate in a web page a number of properties that are useful for identifying a person, including data like the date of birth, phone, occupations or works produced during lifetime. A detailed overview of the task is presented in [3]. Attribute extraction was identified as a useful preprocessing step for the clustering task that started with the previous WePS challenge [1].

Personal attribute extraction is an interesting task with several applications to the more general task of searching people on the web. The hypothesis that attributes could help in the automatic clustering of pages for people with the same name need to be further researched. Nevertheless, attribute extraction is a desirable enhancement for people search interfaces. Photos, but also known characteristic attributes are the main clues that a user have when finding or re-finding people on the web. The use of structured information in search results snippets has been deployed in general search engines like Yahoo! SearchMonkey[1], web people search engines like Pipl[2] or Spock[3]. It also enables the exploration of other interesting interfaces for result navigation like faceted search, map and timeline visualization.

The UC3M team submitted five different configurations of a base system for evaluation to the WePS-AE task. The system has two susbsystems, an attribute extraction pipeline and a pattern acquisition subsystem which are described in Section 2. Section 3 outlines the configuration of the runs and the results obtained. It also includes discussion of the results and initial experiments to clarify the overall poor performance. Future lines of work that we believe should be explored in the task and other particular improvements for our system are presented in the last section.

2. SYSTEM DESCRIPTION

The UC3M system filters attributes based on lexical patterns that are automatically acquired. The acquisition system is based on a bootstrapping algorithm that for a given NE class builds a graph with extracted NEs and patterns that help to signal the given class. Our aim was to test if we were able to acquire indicative patterns that could be correlated with attributes of a class and that could help to filter correct attribute values for a person from incorrect ones.

Although the system was intended to handle different kinds of attributes, due to time restrictions we decided to focus on evaluating only a subset of the attributes. The selected attributes were those that corresponds to typical NE classes like person, location and organization. Those kind of attributes were the kind that the bootstrapping system was able to handle at the time of the evaluation. We expect to extend the method for the rest of attributes.

2.1 Attribute extraction pipeline

The attribute extraction pipeline processes pages for each name using three steps that are depicted in Figure 1 and explained below:

- **Page Preprocessing** includes page HTML cleaning and sentence splitting. We have used the Jericho HTML

Copyright is held by the author/owner(s).
WWW2009, April 20-24, 2009, Madrid, Spain.
.

[1] http://developer.yahoo.com/searchmonkey
[2] http://www.pipl.com
[3] http://www.spock.com

Attribute	NE	technique	tool	P	R	F	Num	
Date of birth	date	NERC	OpenNLP	1.67	54.41	3.24	178	
Birthplace	location	NERC	OpenNLP	1.04	25.15	2.00	154	+
Other name	person	NERC	OpenNLP	1.37	46.99	2.66	281	+
Occupation	occupation	lists		0.00	0.00	0.00	586	
Affiliation	org	NERC	OpenNLP	2.60	30.06	4.78	529	+
Award	award	lists		0.00	0.00	0.00	78	
School	org	NERC	OpenNLP	3.30	52.09	6.21	146	+
Major	subject	lists		0.00	0.00	0.00	65	
Degree	title	lists		0.00	0.00	0.00	90	
Mentor	person	NERC	OpenNLP	1.27	73.07	2.49	17	+
Nationality	nationality	lists		0.00	0.00	0.00	60	
Relatives	person	NERC	OpenNLP	3.78	61.75	7.13	146	+
Phone	phone	regexp	Java	34.5	56.32	42.79	70	
Fax	phone	regexp	Java	0.00	0.00	0.00	2	
Email	email	regexp	Java	19.0	93.48	31.62	91	
Web site	website	regexp	Java	10.6	87.50	19.01	45	
TOTAL (16)				2.57	27.23	4.70	2538	

Table 1: Correlation between attributes and NE values and recognition in training

Parser[4] for removing all the non-textual content of the crawled pages like scripts, style sheets and headers. HTML tags have also been removed but some of them like P and TD have been substituted by breaklines in order to make easier subsequent processing. Nevertheless, the text contains a great amount of noise and the process need often to be halted for large pages. The extracted text is further processed and splitted into sentences using OpenNLP Sentence Splitter[5]. We use the pre-trained model that is provided in the website.

- **Candidate attribute selection**. The recognition of attributes was thought to use a different technique depending on the type of the attribute. An off-the-self NERC system based on ML techniques like OpenNLP is used to recognize attributes that correspond to basic NE types. We used models for Person, Location, Organization and Date that are trained in news corpora. Attributes like Phone and Email are recognized with regular expressions though they need to be robust to achieve good recall. The recognition of other attributes like Degree could be based on lists. The Table 1 outlines the initial design and those attributes that were treated for the evaluation are marked with a plus sign, basically NERC attributes. The basic idea was to produce a list of candidates with good recall to filter it in the next step.

- **Attribute filtering** based on lexical patterns. Each candidate attribute is selected if their context matches any of the indicative patterns acquired for that attribute. Contexts and patterns are defined as a window of tokens that surrounds the candidate attribute value. Contexts are composed of tokens and their position to the left or to the right of the value. The generation of patterns starts with a context and produce one or more patterns by applying a generalization function which substitutes a token with wildcards. An example of how contexts and patterns are generated is shown in Figure 2. In our current experiments we have used a whitelisting approach, that is, candidate attribute values pass the filter if they have a context that is matched with any of the patterns.

c	w_1	w_2	w_3	meet	his	wife
c	*	w_2	w_3	*	his	wife
c	w_1	*	w_3	meet	*	wife
c	*	*	w_3	*	*	wife
c	w_1	w_2		meet	his	
d	*	w_2		*	*his*	
c	w_1			meet		

Table 2: Generation of patterns for a context to the right of an entity mention

2.2 Pattern learning system

The main objective of our participation was to evaluate if our work on learning classes of NE could be applied for attribute extraction. SPINDEL [2] is a bootstrapping system for NE dictionaries and indicative pattern lists. The process starts by defining a set of predicates or NE classes of interest for the application, that we call the semantic model. For each of these classes a set of seed names and a high recall regular expression must be given to start the bootstrapping process from unannotated text. The algorithm uses seeds to extract frequent patterns that co-occur with different elements of a class. Frequent patterns, in turn. are used as queries that help to locate new names that could belong to the class. A name that is signaled by several patterns will probably be a correct member of the class. The process works iteratively and it builds a graph of names and patterns. Nevertheless, a careful evaluation of the selected names and patterns should be carried out in every step in order to avoid drifting away from the desired concept. As it has been shown in previous bootstrapping algorithms [4, 5] the simultaneous learning of several classes improves results. This technique, often named counter-training, is also used when using a multi-predicate semantic model in SPINDEL

[4] http://jerichohtml.sourceforge.net

[5] http://www.opennlp.org

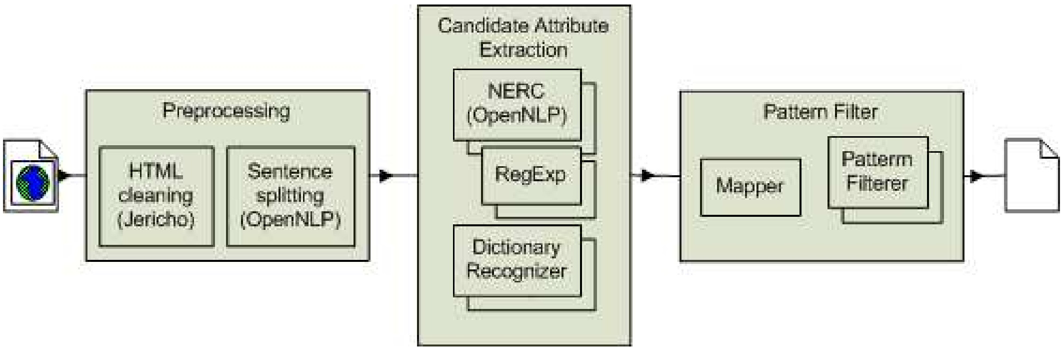

Figure 1: Attribute extraction pipeline

The hypothesis we wanted to test was if acquired patterns were useful to specialize a general NE class into an attribute of the given type. For example given a mechanism to recognize PERSON names, can we use patterns to detect a mention that defines that PERSON as a RELATIVE?.

We have tried different ways to acquire patterns for attribute filtering for each of the runs which are described below. The acquisition step was performed only on the training material. We believe that the bootstrapping process could perform better when much more text is available, but the crawling of more personal pages was not considered due to time restrictions.

2.2.1 Acquiring patterns with positive and negative examples

The first strategy to acquire patterns has used positive and negative examples for each attribute. The WePS-AE training corpus contains a variable number of extracted attributes for each of the pages. For a given attribute like $RELATIVES$, we have selected all the values that has been manually extracted. Then we have used them as positive seeds after removing duplicate values. To collect negative seeds we have extracted all NEs that are valid candidates for that category in the training corpus. For example, for the $RELATIVES$ attribute we have used all the extractions of type $PERSON$ that we found on the training corpus. After removing duplicate values we have also substracted the set of positive instances. Those seeds are used for a predicate $NON\_RELATIVES$. The semantic model is composed of the target attibute and their negation. The process is outlined in Figure 2. We repeated the same process for each attribute. As we have been using only whitelisting, only patterns for the attribute ($RELATIVES$) have been used, but the quality of the patterns is improved by training with positive and negative examples.

2.2.2 Acquiring patterns for several predicates at once

The second strategy to acquire patterns uses only positive examples extracted from the training corpus. In this case, the different attribute values are splitted by attribute type. The semantic model includes all the attribute types. As the system uses the counter-training technique, examples from the rest of predicates help to delimit the current one.

2.2.3 Manual selection of patterns

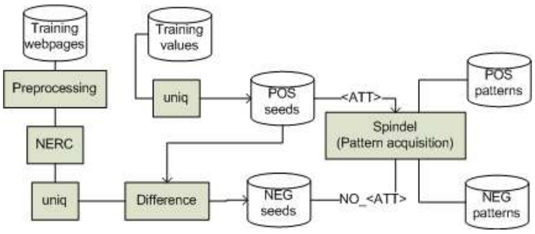

Figure 2: Process for training with positive and negative seeds

The two previous pattern acquisition strategies are completely automatic. The third pattern acquisition strategy combines training with several predicates with manual selection. A human judge cleaned the acquired patterns and removed the noisy ones, those that were not clearly related to the attribute. Another kind of erroneous patterns that were removed were those that were too specific for a person name and that were not properly generalized. The process of cleaning patterns took less than 2 hours. No specific tool was used to help in the process of manual cleaning.

3. RESULTS

We have submitted five different runs using different acquisition methods and also varying some parameters. Runs UC3M_1 to UC3M_3 were submitted on time, while runs UC3M_4 and UC3M_5 were unofficially submitted. Runs UC3M_1 to UC3M_4 are completely automatic but used different methods for pattern acquisition. Runs UC3M_1 and UC3M_3 use positive and negative examples although with different thresholds. The same applies applies to runs UC3M_2 and UC3M_4 which simultaneously learns all predicates. The theshold indicates the minimum support of a pattern τ_p, the number of different attribute values that are required to consider a pattern interesting. The rest of SPINDEL parameters have been the same across runs. The most remarkable one is the maximun length of the patterns ($w = 3$). Run UC3M_5 started with patterns provided by run UC3M_4 but before deploying them in the attribute pipeline were filtered by a human judge. This run obtains the best results among the submitted by our team. Results are summarized in Table 4 where the number in parenthesis indicate the value of the minimum pattern support (τ_p).

Run	P	R	F	training
UC3M_1	2.499	2.177	2.327	pos-neg(2)
UC3M_2	2.401	2.177	2.327	multi(2)
UC3M_3	2.204	1.999	2.097	pos-neg(3)
UC3M_4*	2.204	1.999	2.097	multi(3)
UC3M_5*	**7.953**	**3.643**	**4.998**	UC3M_4 + manual

Table 3: Overall results for the different runs

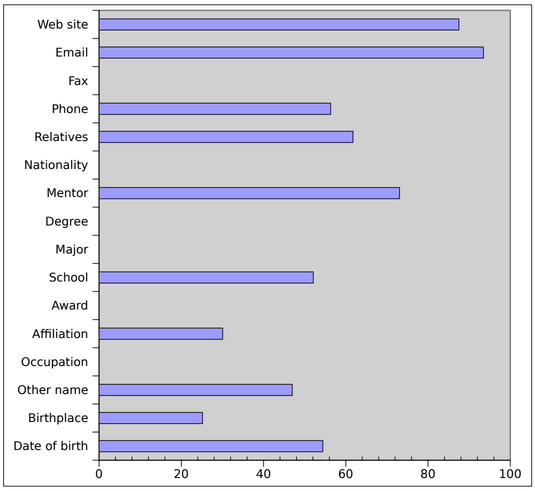

Figure 3: Recall of attribute candidate recognition previous filtering on training data

Attribute	P	R	R
birthplace	17.619	12.375	14.538
affiliation	25.532	2.915	5.230
school	31.532	7.144	11.609
relatives	5.537	27.163	9.199
mentor	0.000	0.000	0.000
othername	0.000	0.000	0.000
all 6	7.953	6.922	7.402
all 16	7.953	3.643	4.998

Table 4: Results analyzed by attribute for run UC3M_5

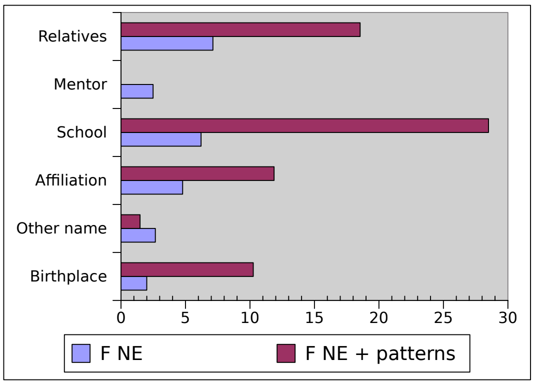

Figure 4: F-measure results for training data

Despite the differences in training methods and parameters, all automatic runs obtain poor results. Almost no difference is observed at this level of performance for training methods. We believe that it is due to the small size of the bootstrapping collection (web pages for 30 WePS-1 names). Manual selection of the patterns does in fact help, particularly to get an increase in the precision of the extracted values. Nevertheless the performance is too low for this run too. Table 4 shows results for run UC3M_5 splitted by attribute. Attributes with a relatively large number of examples like $RELATIVES$ acquire a good number of patterns which help to obtain good recall. For $AFFILIATION$ and $SCHOOL$, precise patterns were acquired but coverage is still unsufficient. Attributes like $MENTOR$ and $OTHERNAME$ had few training examples and no useful patterns were acquired. No special heuristics were applied for the $OTHERNAME$ attribute. Overall recall improves slightly when we consider only the six attributes that we have worked with. We believe that using the current system for the 18 attributes would produce a performance similar to the rest of the systems.

Table 1 and Figure 3 show the performance of the candidate attribute extraction module in training data. The last step, pattern filtering, was removed to produce these results. Recall at this step is an upper bound on the overall system performance using the actual architecture. A large percentage of NEs are not correctly recognized and therefore they can not be proposed as attribute values. Recall performance varies from 25% to 70% which is much lower than NERC tools in news text. Even regular expressions that were deliberately designed for recall do not achieve 90% performance. Further analysis is needed to quantify the errors due to disagreement with the annotation guidelines. It seems that when using the NERC tool directly in webpages, not only precision is affected, but also recall.

Figure 4 in turns show the improvement in F-measure by using pattern whitelisting for the attributes that we completed the whole process. Despite the improvement in most attributes, recall is not mantained at the necessary level.

Finally, we would like to provide some qualitative comments in some of the quirks of the task that make it even more challenging than thought. In our system we have assumed that pages are personal or mainly talking about one person. This assumption do not hold often. Two are the main reasons, the tail of the results often just cite the person and some pages contain information about several people (for example, obituaries). Another pecualirity is the frequent use of coordination structures to express multiple attributes like in . . . *has two daughters Angie (6) and Carol (4)* . . .) , which in our case make short patterns useless.

4. CONCLUSIONS AND FUTURE WORK

The task of attribute extraction has several useful applications in the context of web people search, but it has shown definitely difficult to adapt actual technology and tools. Several issues need to be considered at our system level while some questions need to be addressed at a wider level.

For example, NE tools and other basic NLP tools have not been widely used and evaluated in the context of unrestricted web pages. Deeper work on NERC in noisy environments need to be carried as well as tools that comprise a wider hierarchy of types. People search results will be a mix of semi-structured web pages served from social networks services, pages from Wikipedia with structured content, company or university directories and other heteroheneous personal pages. The combination of wrapper and NLP technologies seems like an interesting research direction in this situation. NE tools need also to provide a larger number of classes in order to support attribute extraction. Finally, regarding the selection of attribute values, an open question is why even manual patterns achieve low precision when signalling attributes which deserves to investigate the characteristic of this text style.

In what concerns our system, there are many lines to explore yet. We would like to explore the use of SPINDEL not only for pattern extraction but for building fine-grained NE recognition with types like *AWARD* or *DEGREE*. In order to effectively acquire list of NE and patterns for those fine-grained classes we need larger corpora whether of personal pages or general ones. There are also other alternatives for the use of patterns that could be explored like using blacklisting and combine patterns as features for supervised machine learning. We are also convinced that additional information like the use of markup, structural patterns and distance heuristics could help to discriminate correct values. Moreover, the integration of a priori knowledge like cardinality or attribute specific check rules would also have an impact in some attributes.

5. ACKNOWLEDGMENTS

The authors would like to thank Mayte Vicente Díez and David del Valle for useful discussion regarding our previous WePS participation in the clustering task. This research work has been supported by the Regional Government of Madrid under the Research Network MAVIR (S-0505/TIC-0267) and by the Spanish Ministry of Education under the project BRAVO (TIN2007-67407-C03-01).

6. REFERENCES

[1] J. Artiles, S. Sekine, and J. Gonzalo. Web people search: results of the first evaluation and the plan for the second. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1071–1072, New York, NY, USA, 2008. ACM.

[2] C. de Pablo-Sánchez and P. Martínez. Building a graph of names and contextual patterns for named entity classification. In *31st European Conference on Information Retrieval*, 2009.

[3] S. Sekine and J. 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*, 2009.

[4] M. Thelen and E. Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 214–221, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[5] R. Yangarber. Counter-training in discovery of semantic patterns. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 343–350, Morristown, NJ, USA, 2003. Association for Computational Linguistics.