WePS2 Attribute Extraction Task

Satoshi Sekine
New York University
715 Broadway, 7th floor
New York, NY 10003 USA
+1-212-998-3175
sekine@cs.nyu.edu

Javier Artiles
UNED NLP&IR group
Ciudad Universitaria, s. n.
28040 Madrid , Spain
+34 91 398 8106
javart@bec.uned.es

ABSTRACT
In this paper, we describe the Web People Search 2 attribute extraction task (WePS2-AE). It was conducted in September-December 2008 along with the WePS2 clustering task. Six groups participated in the AE task. We will describe the motivation, task definition, evaluation set up, participating systems, and evaluation results. We will discuss the problems and future directions.

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
Performance, Design, Experimentation, Security, Human Factors, Languages.

Keywords
Information extraction, Attribute extraction, Disambiguation, person names, Web documents

1. INTRODUCTION
At the evaluation of WePS1 [1], we identified a new challenge of "attribute extraction" for people on the Web pages. It was noticed by the participating systems and annotators that attributes, such as birth date, spouse name, occupation and so on, are very important clues for disambiguation. We believe it is the right direction to study such problem and try to implement technologies to identify such attributes. We made this evaluation an independent subtask at WePS2, along with the clustering task [2].

In order to set up the task, the first challenge is to define what are "the attributes of people". These have to be general enough to cover most people, useful for disambiguation, and meaningful for the evaluation. We took an empirical approach to define them; we extracted possible attributes from the Web pages and created a set of attributes which are frequent and important enough for the evaluation. We looked at 156 documents from the WePS corpus, and annotators extracted as many attribute-value pairs as possible. The annotators are instructed to extract attributes of people which can be expressed as "Person's *Attribute* is *Value*". The attribute and the value must be a noun or its equivalent. An attribute and value pair may be expressed in a tabular format, or only the value may be mentioned in a sentence. If the name of the attribute is not explicit in the web page (e.g. "I am a professor" means Person's occupation is professor), then the annotator creates the attribute name. From the 156 documents, the annotators found 123 kinds of attributes; the 6 most frequent attributes are Occupation (116), Work (70), Affiliation (70), Full name (55), Person to work with (41) and Alma Mater (41). The number in parenthesis is the number of pages in which the information was mentioned. Among the 123 attributes, there are attributes which are not suitable for the evaluation. For example, domain dependent attributes, such as "Career Points for a basketball player", or an attribute of an attribute value, such as "Birthday of spouse" are not suitable for the evaluation. Also, there are a set of attributes which might be meaningful even if we merge them together, such as "father", "mother", "sibling" as "relatives". By selecting and merging the 123 attributes, we finally made up 18 attribute classes, as shown in Table 1. Note that for the training data, we set up 16 attributes. However, between the training and test, we decided that "education", one of the 16 attributes, should be divided into three attributes "school", "major" and "degree". We recognized that the change would affect systems which use supervised methods, but the change was made mainly because the values of those three types of attributes are actually different types of entities.

The subtask involves extracting the values of those attributes as accurately as possible from Web pages. It would be ideal if we can merge the information for a single person entity from multiple pages, but if we set up such an evaluation, it is not easy to handle the effect of clustering mistakes. So, we evaluated the result for a given page, based on precision, recall and F-measure.

As we expected, the problem was solved by a combination of many technologies, such as named entity recognition and classification, text mining, pattern matching, relation discovery, information extraction and more. Conducting this evaluation will provide a good opportunity to develop and collect useful resources, such as lists of named entities, occupation names and so on, annotation tools or text mining tools. We hope that this evaluation will provide fundamental research opportunities, as well as practical industrial application opportunities in the near future.

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

2. TASK DEFINITION

The formal definition is described in a 10-page definition document, available at the following URL.

http://nlp.uned.es/weps/weps2/WePS2_Attribute_Extraction.pdf

In this paper, the overview and summary of the definition will be presented.

2.1 Overview

This subtask is to extract 18 kinds of "attribute values" of target individuals whose names appear on each of the provided Web pages. The organizer will distribute the target Web pages in their original format, (i.e., html), and the participants will be expected to extract attribute values from each page. The individual name associated with a particular page will be given, and the attribute values for that person should be extracted. Web pages containing multiple individuals sharing the same name will NOT be given. All attributes to be extracted are listed in Table 1 below. Although there are 18 attributes listed, "Work" and "Location" will NOT be evaluated for WePS-2, since after annotating the 300 sample texts, those attributes were found to be very ambiguous due to the degree of variation among the target individuals.

Table 1. Definition of 18 attributes of Person at WePS2-AE

	Attribute Class	Examples of Attribute Value
1	Date of birth	4 February 1888
2	Birth place	Brookline, Massachusetts
3	Other name	JFK
4	Occupation	Politician
5	Affiliation	University of California, Los Angels
6	Work	The Secrets of Doroon
7	Award	Pulitzer Prize
8	School	Stanford University
9	Major	Mathematics
10	Degree	Ph.D.
11	Mentor	Tony Visconti
12	Location	London
13	Nationality	American
14	Relatives	Jacqueline Bouvier
15	Phone	+1 (111) 111-1111
16	FAX	(111) 111-1111
17	Email	xxx@yyy.com
18	Web site	http://nlp.cs.nyu.edu

2.2 General Rules

a) Attribute values should only be extracted from the pages provided. Those should be extracted AS IS. Attribute values which don't exist in the given pages should not be extracted. Do not extract a value from any pages that are linked from the pages provided.

b) If there are two or more attribute values for one attribute class, participants should extract all the values. For example, both "Japan" and "Tokyo" can be extracted as values of "Birthplace" from the expression, "He was born in Japan, in the city of Tokyo." However, if the two values are used in a single phrase, they must be extracted as one value. For example, the entire phrase "Tokyo, Japan" must be extracted from the expression, "He was born in Tokyo, Japan."

c) An expression should be extracted even if it contains a factual error. For example, both "1782" and "June 25, 1841" should be extracted as values for "Date of Birth" from the following sentence: "Macomb, Alexander (1782-1841) General: Alexander Macomb was born on Detroit, Michigan, on June 25, 1841."

d) If a single attribute value is expressed in more than one way on the same page, each expression should be extracted as a separate value.

e) Do not extract a value written in a non-English language.

f) If a single attribute value appears differently because of spacing and/or punctuation (i.e., capitalization at the beginning of a sentence), it is not necessary to extract each expression. No penalty or advantage will be given to participants who produce the same value multiple times.

g) If there is a line break in an attribute value, the break and spaces adjacent to the break can be replaced by a single space. No penalty or advantage will be given either way. The evaluation program will replace a sequence of spaces and breaks by a single space before the evaluation.

h) The determiner ("the") at the beginning of a name is optional in the evaluation. No penalty or advantage will be given if the determiner is included or omitted.

i) Non-ASCII characters are treated specially. In the answer files, these characters are replaced by "?" character. No penalty or advantage will be given if no match will be made with those characters.

2.3 Pages to be ignored

The following pages will not be used in the evaluation.

a) A page that does not contain the exact string of the name of a target person. For example, if the target name is "John Kennedy," and the name appears on a given page as "John F. Kennedy" only, the page will not be used.

b) A page that has two or more individuals sharing the same target name. For example, a page which contains "John Kennedy (Politician)" and "John Kennedy (Actor)" will not be used.

c) A page that displays search results from databases (e.g., DBLP and CiteSeer) or shopping sites (e.g., amazon.com and Yahoo! Shopping).

d) A page that is not written primarily in English.

e) A page on which the target name refers to a fictional character.

f) A page with fictional content, even if the target person in the fiction is a real-life figure.

3. EVALUATION

The evaluation of attribute values for WePS2 was conducted on the same data used for the clustering task on WePS2. In this section, we will first briefly describe the data. Then we will

describe the annotation scheme, statistics of the data, and the evaluation scheme of the task.

3.1 Data

The WePS-2 test data consists of 30 datasets, and three different sources where used to obtain the names, namely, Wikipedia, ACL'08 and US Census. For each name we obtained the top 150 search results from an Internet search engine (Yahoo! API). This dataset was used for the clustering task, as well. We provided a training data consisted of 17 person names from the WePS-1 data.

3.2 Annotation

The attribute data was annotated by 4 independent annotators. Three of them have a background in linguistics with a linguistics master's degree, and the other is primarily a computational linguist, having worked in the field for more than three years. The main annotator, who has helped in the definition, initial trials, and annotation of all the training data, leads the entire annotation effort and provides consultation for the other annotators. He annotated most of the test data and did the adjudication for the final decisions. So, the annotation strongly reflects his understanding of the task. The agreement with the other annotators ranges from 60%-85%.

3.3 Statistics

In this subsection, we will present some statistics of the data.

First, Table 2 shows the statistics of the number of documents. In the WePS2 test data for 30 people names, there are 3,468 Web documents (on average 115.6 documents per name). We ignored 585 documents for the reasons given in section 3.2. Out of 2,883 documents used for the test, 2,421 documents have at least one attribute value while 462 documents have no attribute value.

Table 2. Statistics of the documents

Number of documents in the test data	3,468
Number of documents ignored	585
Number of documents used for the test	2,883
Number of documents with at least one attribute value	2,421
Number of documents with no attribute value	462

Table 3 shows statistics of the attribute value, the total number of attribute values in the entire test data, the average number of attribute values in a single document of the test data used (2,883) and the maximum number of attribute values appearing in a single document. As you can see, the attributes with the largest number of values are "Work" and "Affiliation", but "Work" has a large variation. This was observed in the test data, because there are some types of people, such as artists and novelists, who have a lot of work titles. Many of the attributes which you may think of as a common attribute, such as date of birth, birth place, nationality, phone, email and web site, have an average of 0.1. It means that those attributes appeared once in 10 documents on average. This is a bit surprisingly small, considering that some of the names we selected are names of scientists and famous people.

Table 3. Frequency of Attribute values

	Attribute Class	Total Number	Average per doc	Max. per doc
1	Date of birth	370	0.12	4
2	Birth place	301	0.10	4
3	Other name	797	0.27	6
4	Occupation	3,292	1.10	20
5	Affiliation	3,105	1.03	19
6	Work	3,770	1.25	141
7	Award	264	0.09	14
8	School	494	0.16	10
9	Major	173	0.06	6
10	Degree	335	0.11	6
11	Mentor	343	0.11	12
12	Location	888	0.30	8
13	Nationality	250	0.08	2
14	Relatives	914	0.30	29
15	Phone	219	0.07	5
16	FAX	65	0.02	2
17	Email	209	0.07	5
18	Web site	154	0.05	4

3.4 Evaluation Scheme

Evaluation will be conducted by comparing the system output and gold standard data created by annotators. The gold standard data will be created by annotators before consulting the system output. Then, all the spurious values generated by the systems may be checked by annotators to see if there are answers missing from the key. Such a scheme has been chosen due to the success of the pooling scheme in the field of information retrieval.

The comparison will be done using recall, precision and F-measures for each individual attribute and for the overall answers.

Recall = (# of correctly identified attribute values by system) / (# of attribute values in gold data)

Precision = (# of correctly identified attribute values by system) / (# of attribute values the system produced)

F = 2 * Recall * Precision / (Recall + Precision)

4. SYSTEMS

Six groups have participated to the evaluation. These are PolyUHK, ECNU (2 systems), MIVTU, CASIANED, UC3M (5 systems) and UvA (5 syetems) in the random order. In this section, a brief description of each system will be presented. The detailed description will be presented in the papers of this workshop.

4.1 PolyUHK

The AE system is rule-based, and tries to catch some typical patterns in Webpage. Unlike most previous pattern learning, which limits to a sentence, our patterns often are learned from continuous sentences. For example, the typical occupation and affiliation pattern as follows.

Alexander Macomb
Captain, United States Navy

4.2 ECNU

We use processing pipelines with multiple techniques including web content and structure cleaning, NER, regular expression patterns, and so on.

METHODOLOGY

1. Web Page Cleaning

Like most commonly-used HTML cleaner, we first do a shallow cleaning as follows:

- repair missing or non-closed tags; strip away all HTML tags, script codes, CSS codes as much as possible; and then the content between the title, the body and the anchor tags is extracted for each page.

However, the resulting documents still contain a large amount of noises. Thus we do a further deep cleaning as follows:

- some HTML tags are replaced by white space (such as <p>, <td>), while others are converted into line separators (such as <li>,
, <tr>); remove content between a pair of tags and controllers, such as <select>, <style>, ListView, ListBox, ComboBox, etc.; remove all non-*url* content between a pair of anchors; remove all textual content after "copyright" keywords, etc.

2. Target People Attribute Extraction

For different kinds of attribute, there are different ways. Regarding four enumerable attributes, i.e., *Occupation*, *Major*, *Degree*, *Nationality*, a dictionary (list) is constructed from public resources (such as *wikipedia*). Then we use a simple dictionary matching algorithm to extract these attributes values. With regard to the 7 attributes, i.e., *Birth place*, *Affliliation*, *School*, *Mentor*, *Relatives*, *Email*, *Web site*, we first adopt a NER tool named ESpotter [?] to extract locations, names, organizations, emails and urls from web pages. Then for each attribute, we examine if the corresponding attribute trigger keywords are available in the same sentences. Specifically, for *Email* and *Web site*, we also construct a stop list for filtering, including the very common values such as "webmaster@xxx", "wikipedia", "wiki", and so on. For *Date of birth*, we ¯rst construct several types of regular expressions to recognize textual or numerical date. Then we check if current sentence has trigger keywords, such as \born", \birthday", \birth date", \birth", etc., and has a referent for the target person as well.

With respect to *Other name* attribute, we generate several name regular expression patterns, such as first name alone, last name alone, capitalized first letter from the first name or last name in combination with capitalized letters and the names, etc.

For *Phone* and *FAX* attributes, it is easy to extract them by using regular expression patterns. Then we further identify if it is a phone or fax number by examining the relative locations of trigger keywords in the sentence. Extracting *Award* attribute is the most challenging task due to the great diversity of expressions in natural language.

We explore a lot of methods and no one performs well.

4.3 MIVTU

The proposed method is illustrated in Figure 1 and it can be seen as consisting of two fundamental steps. First, we mark potential attribute values in a given text. Second, we decide which candidate values correspond to which attributes of the given person name.

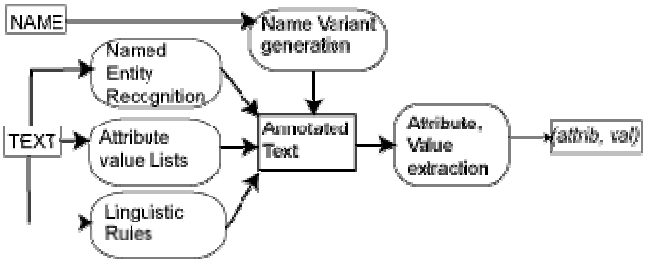

Figure 1. MIVTU system diagram

To mark the potential values of attributes we use three approaches: lists of candidate attribute values, a named entity recognizer, and a set of manually created rules in the form of regular expressions. For example, attributes such as nationalities (e.g. Japanese, British), universities (e.g. The University of Tokyo), majors (e.g. Master of Science, Bachelor of Arts) and professional titles (e.g. professor, general) can be marked using candidate lists. These lists were created manually referring online information sources such as Wikipeida. However, lists cannot completely enumerate all attribute values. In addition to using pre-compiled lists of attribute values, we used a named entity recognition tool to mark three types of named entities: personal names, organization names, and location names. Attributes such as dates, telephone numbers, fax numbers, e-mail addresses and urls usually follow a fixed format and can be efficiently annotate in a text using rules in the form of regular expressions.

Once the given text is annotated following the above mentioned procedure, we mark all potential variants of the given name for which we must extract attribute values. We generate abbreviated forms, last name and first name inter-changed forms, middle name initialized forms, middle name dropped forms, name followed by titles, and combinations of all the above. We then mark those variants in the given text. For example, if the given name is *John Fitzgereld Kennedy* then this process will generate variants such as *J. F. Kennedy*, *John F. Kennedy*, *Kennedy J. F.*, and *John Kennedy*. To find the attributes of the given person, we find the distance for each marked attribute value from a name variant. We then select the closest attribute value as the correct candidate. However, we do not go beyond a different person name when computing distances. Moreover, we assign higher confidence score to an extracted attribute value if certain cue phrases appear in close proximity. For example, the cue phrases *born* and *birth* increase the confidence of an extracted date being the date of birth of the person under consideration. Likewise, cue phrases *mentor*, *supervisor*, and *advisor* increase the confidence of a value extracted as the mentor of the person under consideration. The cue phrases are selected manually after reviewing the test data in the WePS attribute extraction dataset. Each sub-component of the proposed attribute extraction system including examples of candidate value lists, linguistics rules, cue phrases, and attribute extraction method will be further explained in the sections to follow.

4.4 CASIANED

In this section, we describe the CASIANED's people attribute extraction system. A graphical diagram presenting our method is shown in Figure 2.

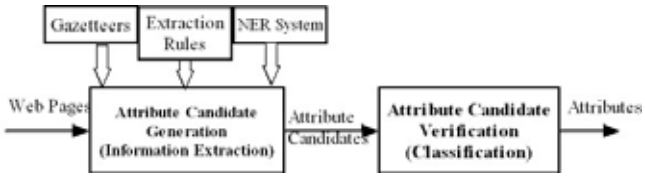

Figure 2. CASIANED system diagram

The method is essentially composed of two function modules: the attribute candidate generation module and the attribute candidate verification module. For a web page, the attribute candidate generation module extracts all attribute candidates for every attribute class through recognizing different named entities and noun phases which can be used as attributes, then the attribute candidate verification module verifies these candidates through classification.

1 Attribute Candidate Generation: The attribute candidate generation module extracts all attribute candidates in a given web page through information extraction. For a given attribute class, the attribute value usually is noun phrase of some special types. For example, the value of Date of birth attribute must be date; the value of Birth place must be location, etc. We category the attribute classes of people into three different categories according to their value type, as shown in Table 1. For different attribute value types, special methods are used to extract attribute candidates.

Table 4. The Value Type of Different Attribute Class and the Corresponding Candidate Extraction Method

Value Type	Detailed Value Type	Attribute Class	Candidate Extraction Method
Traditional Named Entity	People Location Organization Date	Date of birth, Birth place, Other name, Affiliation, School, Mentor, Relatives	Named Entity Recognition tools
Special Type Named Entity	Email Phone Number URL	Award, Nationality, Phone, FAX, Email, Web site	Extraction Rule (Regular Expressions)
Special Type Noun phrase	Occupation Degree Major Award Nationality	Occupation, Work, Major, Degree	Gazetteer based matching

The first value type of some attribute classes is the traditional named entities (People, etc.), for example, the value type of Date of birth is date, the value type of Affiliation, School is organization, etc. The existing approach has been able to achieve good results in traditional named entity recognition (Nadeau, David et al. 2007). We extract the named entities from the web page using the OpenNLP (http://opennlp.sourceforge.net/) Named Entity Detection tools. Then the extracted named entities are considered to be the candidates of corresponding attribute classes. The second value type of some attribute classes is the named entity of special types. Currently, there is no tool which can be used to extract such named entities instantly. Fortunately, there are some patterns existed on Email, Phone Number and URL. We can extract them using some extraction rules. In our system, we build several regular expressions to extract Email, Phone Number and URL from Web pages. The third value type is noun phase of specific types. For example, the noun phase about occupation, such as professor, artist, actor, etc. We use a gazetteer based matching method to extract the attribute candidates for this value type. For each noun phase type, we build a gazetteer from the Wikipedia2, the largest online encyclopedia which contains millions of concepts. For example, we build the occupation gazetteer through extracting all occupations listed in the List of occupations entry in Wikipedia. Based on these gazetteers, we can extract the attribute candidates by finding all word appearances of the corresponding gazetteer in Web pages. After the attribute candidate generation, a list of attribute candidates is obtained for every attribute class.

2 Attribute Candidate Verification through Classification: The attribute candidates obtained using the method described in previous section need to be verified. This is because a candidate may be the attribute of other persons or not attribute at all. For example, a phone number candidate may not the phone of the target individual, or it is actually a Fax. Our system verifies the attribute candidates through classification. For every attribute class, a classifier is trained to identify whether an attribute candidate is the real attribute of the target individual. For a given person and a given attribute class, the classifier classifies an attribute candidate into two targets: attribute or not attribute. We build the training corpus for each attribute class's classifier as follows: firstly, we generate all attribute candidates of the given attribute class in the develop dataset, where an attribute candidate is represented as (value, context text); secondly, we label the attribute candidates whose value can match one true attribute as positive training instance, and all left candidates are taken as negative training instances. We need to extract the representation of an attribute candidate as the input of the classification process. An attribute candidate is represented as a vector of features as follows. *Context Token*. The context words near an attribute candidate usually provide helpful information. For example, the word Fax within Fax +43 (1) 58801 – 18392 provides the evidence that +43 (1) 58801 – 18392 is a Fax. We firstly extract the words within a window size 5 as context words, then stem them using the Porter stemmer3, stop words are filtered. Each word retained is used as a feature. *Value Pattern*. For a specific attribute class, the value usually shows some patterns. For example, some patterns existed in the School attribute value, such as School_of_Location, Location_State_College, etc. We extract the pattern features of value through the following steps: firstly we recognize and label the named entity within the attribute candidate, for University of Cape Town the label result is University of <LOCATION>Cape Town<LOCATION>; then we replace the named entity with the label, for the above example, the result after replacing is University of LOCATION; then we extract all the unigrams, bigrams and the total strings as pattern features, we also filter the features which are too general. For the

above example, the extracted features are University, University of, University of LOCATION. *Dependency Path.* We model the relation between the target individual and the attribute candidate by analyzing the dependency path between them. For example, one dependency path between Christine Borgman and the Occupation candidate professor is “the Christine Borgman, a professor at UCLA”, which provides evidence about the Christine Borgman is a professor. We extract the features on dependency path through the following steps: firstly we label the dependency path with POS tag and NER tag; then we extract all unigrams, bigrams and trigrams, all grams extracted are used as features. We also add the number of named entity on the dependency path and the length of dependency path as features. Using the features described above, we can classify whether an attribute candidate is the specific attribute class’ value of the target individual. After the attribute candidate verification, all attribute candidates retained will be taken as real attributes.

4.5 UC3M

The UC3M system filters attributes based on acquired lexical patterns. We have focus only on a subset of the attributes, those that were clearly NEs and that our pattern learning system was able to handle at that moment.

Each page was processed following three steps:

1. Preprocessing. HTML tags and scripts are stripped and extracted text is splitted in sentences. We have used Jericho HTML Parser and OpenNLP for each task.
2. NE filter. A NERC signals four different types of entities (Person, Location, Organization, Date) that are selected as candidates for an attribute. For example, entities tagged as persons are candidates for the Relatives attribute. NEs are recognized using OpenNLP based on off-the-self Maximum Entropy Models trained on tagged news texts.
3. Pattern filter. Each candidate attribute is selected if their context matches any of the indicative patterns acquired for the attribute. The contexts and the patterns are characterized as a window of tokens that surround the NE.

The process to acquire patterns for each attributes is executed off-line and it proceeds by bootstrapping the patterns from the training material. We have experimented with different strategies using SPINDEL[1]. The first strategy extracts frequent patterns using positive and negative seeds. Negative seeds are NEs of the same type that have not been selected as attributes. The second strategy bootstraps simultaneously patterns for different attributes. Finally, some experiments have use manually filtered patterns.

4.6 UvA

Our aim for the attribute extraction task was to experiment with two fundamentally different families of methods: manual and automatic pattern construction. Both are applied on top of a named entity tagged version of the documents. Under the first group of approaches, a separate extraction strategy was developed for each attribute type. We submitted two runs, using a baseline UvA_1 and an advanced UvA_2 version of these patterns, as follows:

Date of birth Dates are extracted using simple regular expressions. For the baseline all matching patterns are returned as dates of birth. Advanced only returns dates if they are contained within brackets, contain the words ``birth" or ``born", or the first and last name of the person close to it.

Birth place The output of the named entity recognizer is used, and all locations are selected as birth place. The advanced version filters these locations based on the date of birth: only locations following this (advanced) date are selected.

Other name Regular expressions are used to select capitalized words (e.g. names) or abbreviations (e.g. initials) before, after or in between the first and last name of the person. In the advanced setting, the candidate names are filtered against a list of 2,600 names, to prevent regular words ending up being ``other names".

Occupation We compiled two gazetteers of words and phrases describing occupations: one using Wordnet and another using Wikipedia. To extract possible occupations, we simply identified all occurrences of terms from our gazetteer. The baseline run used the gazetteer created using Wordnet. The advanced run used both Wordnet- and Wikipedia-based gazetteers.

Affiliation As affiliations of a person, we simply extracted all organizations (as identified by a named entity tagger) from the sentences containing the person's name.

Award Our baseline run uses the pattern ``the ... AWARDTYPE for$|$the ...", where AWARDTYPE is either ``Award", ``Prize", ``Competition" or ``Medal". The advanced approach extends it by adding more award types (for example ``Fellowship", ``Hall of Fame", ``Honour") and advanced patterns for extracting time-dependent awards (e.g., ``The Author of the Year").

School Baseline method extracts schools that follow simple patterns like ``University of ..." and ``... College"; the advanced setting looks for occurrences of Degrees and Majors and follows more sophisticated patterns to recognize schools.

Major The baseline uses the pattern ``degree in.." to detect majors. For the advanced run we use the degrees extracted in the advanced degree setting.

Degree In the baseline setting we select all degrees from list of 89 degrees (and 110 abbreviated versions). The advanced run uses the same list, but removes degrees if no major or school is present in the documents (as a post-processing step).

Mentor Any sequence starting with ``influence", ``with", ``by", ``for" followed by at least two words with their first letter capitalized; advanced is the same as baseline.

Nationality In the baseline, we use a list of nationalities and select all we can find. The advanced run uses the same list, but filters on closeness of the first or last name of the person.

Relatives A dictionary of terms describing family relations (father, mother, grandson, etc.) was constructed manually. Baseline returns the first person name occurrence after a relation term. Advanced uses a refined dictionary of relations and returns all names within a fixed window size around relation terms.

Phone and Fax are extracted using regular expressions (no advanced version). All number matching the patterns are used as both telephone and fax number.

Email address and Web site are extracted using regular expressions. Baseline returns any email/url identified, advanced returns only those that contain either the first or the last name of the person.

In runs UvA_3 to UvA_5 we apply an automatic approach for ``learning" patterns from training examples. We experiment with two different ways to construct patterns and with using named entity tags as an additional source of information.

We use a two-step approach. In the first step, we extract candidate patterns using the supplied sample data. For each name in the test set, and for each attribute, we extract from the text of the supplied web pages each sentence that contains the person's last name and an attribute value. We then construct patterns for 1 up to 5 tokens that precede and follow the attribute value in that sentence. Next, we generalize patterns by replacing elements of

person names by placeholders, and, when using NE-tags, by additionally replacing named entities by their tag.

In the second step, we select patterns following a leave-one-out approach. The patterns extracted from all but one person name are applied to extract attributes from the web pages belonging to the held-out person name. Thus, we evaluate the extracted attribute values and keep the patterns where F-score exceeds a threshold value. This is repeated until each person name has been held-out once and the resulting set of patterns is applied to extract attributes for unseen names.

Experimental results showed that our manual patterns are very effective, in terms of recall. Further, the refined versions significantly improve precision, without losing any of the recall. As to the automatic methods, we conclude that it is possible to automatically learn patterns for attribute extraction. However, performance of the approach used strongly varies with the attribute type. There are several interesting directions for future work.

5. EVALUATION RESULTS

In this section, we will present the evaluation results. It is our belief that the evaluation figures for each system are not very important at this stage, as each individual system has its own characteristics. We will focus more on the analysis of the task and statistics in general.

5.1 Results by System

Table 5 shows the precision, recall and F-measure of each system. The scores are generally very low. It means that the task has a lot of challenges and it is important to see what the difficulties are and to give some statistics which may suggest possible solutions.

Table 5. Results by System

System	Precision	Recall	F-measure
PolyUHK	30.4	7.6	12.2
ECNU_1	6.8	18.8	10.0
ECNU_2	8.0	17.6	11.0
MIVTU	5.7	15.5	8.3
CASIANED	8.5	19.0	11.7
UC3M_1	2.5	2.2	2.3
UC3M_2	2.4	2.2	2.3
UC3M_3	2.2	2.0	2.1
UC3M_4*	2.2	2.0	2.1
UC3M_5*	8.0	3.6	5.0
UvA_1	2.7	27.3	5.0
UvA_2	4.4	27.4	7.6
UvA_3	0.7	0.2	0.2
UvA_5	0.2	0.0	0.0
UvA_5	3.3	2.8	3.1

* indicates unofficial runs, which are sent after the deadline

Table 6 shows the number of matched, spurious, and missed attribute values for each system. Only one (best) system from each group will be used in the analyses below. We can see how many attribute values are detected by systems out of 11,253 attribute values and how many are wrongly generated. There are great variations, but it should be noted that only 3,085 correct attribute values can be detected by the best recall system. We believe we have to figure out how to extend the coverage, because otherwise this missing information can never been detected. Also, we should have filtering technologies which reduce the number of spurious candidates.

Table 6. Number of Match, Ovg, Miss by System

System	F-measure	Matched	Spurious	Missed
PolyUHK	12.2	860	1,966	10,393
ECNU_2	11.0	1,977	22,682	9,276
MIVTU	8.3	1,743	29,064	9,510
CASIANED	11.7	2,138	23,032	9,115
UC3M_5	5.0	410	4,745	10,843
UvA_2	7.6	3,085	66,762	8,168

5.2 Results by Attribute Type

Figure 3 shows the F-measure result by attribute type in the order of the best system performance. The easiest attributes are “email”, “FAX”, “birthplace”, “degrees”, “date of birth” and “phone”. It is mix of traditional Named-Entity type attributes (“birthplace” = location, “date of birth” = date), types of attributes which have very typical format (“email”, “fax”, “phone”) and limited vocabulary attributes (“degree”). On the other hand, the difficult attributes are “mentor”, “relative”, “occupation”, “affiliation” and “award”. These are also a mix of traditional NE type attributes where disambiguation is needed (“mentor”, “occupation” and “affiliation”), and unfamiliar NE types (“occupation” and “award”). The first type has to be solved by context, such as direct indicators of the type of person in a table, or the surrounding context in sentences which indicate the relationship of the person to the main person. This is a complicated problem, provided the training data is not so large. We may want to use a bootstrapping method to gather such context clues. The second problem, unfamiliar NE type, should be solved by introducing such NE types. For example, the 200 extended NE categories [4] includes “occupation” and “awards”, however, the accuracy of the tagger and the list in the dictionary have to be improved.

5.3 Results by People Name

Figure 4 shows the F-measure results by people name. We can see that the results vary greatly from name to name. You may notice that for some names such as Mirella_Lapata, Janelle_Lee or Louis_Lowe, the performance of the PolyUHK system is outstanding even though the other system’s performances are very low. This is probably due to the fact that the PolyUHK system is a high precision system (Table 5), and those names have a relatively small number of attributes. I believe the conservative strategy for those names makes PolyUHK’s performance higher. Other than those names, the system performance for different names are relatively similar. It suggests that the technologies the participants have used in this evaluation are not very different.

6. ANALYSES

In order to see the real problems and to find possible solutions, we conducted a few specific analyses on the results.

6.1 Coverage by the UNION system

First, we observe the performance in combination of all the systems. We merge the outputs of all the systems (6 systems) and see the recall and precision of the union system (Table 7). It is natural that the recall becomes much higher than individual systems and precision becomes lower. However, it is very interesting that most of the categories achieve much higher recall than the best individual system, and the best systems are different for different attributes.

Table 7. Performance by the union system

	Attribute Class	**Recall**	**Precision**	**Best recall by individual system**
	All	53.9	2.7	27.4 (UvA_2)
1	Date of birth	74.8	2.2	32.0 (MIVTU)
2	Birth place	69.9	0.5	48.5 (MIVTU)
3	Other name	38.4	2.1	30.2 (PolyUHK)
4	Occupation	56.5	2.1	38.3 (UvA_2)
5	Affiliation	44.6	3.9	23.0 (MIVTU)
7	Award	29.9	2.3	16.7 (UvA_2)
8	School	60.2	4.6	43.5 (ECNU_2)
9	Major	34.1	3.2	16.8 (UvA_2)
10	Degree	65.4	10.9	42.7 (CASIANED)
11	Mentor	18.7	0.3	16.9 (UvA_2)
13	Nationality	81.2	3.8	42.0 (CASIANED)
14	Relatives	61.9	2.7	55.0 (UvA_2)
15	Phone	84.9	10.2	74.4 (UvA_2)
16	FAX	86.2	4.3	70.8 (UvA_2)
17	Email	88.0	13.9	69.9 (ECNU_2)
18	Web site	83.1	4.5	40.3 (ECNU_2)

6.2 Attribute types

We analyzed the errors of the union system. We found that we can categorize the attributes, based on the types of errors, into at least four categories. Table 8 shows the four categories in the order of ease of identification of attribute values. Because they have the different type of errors, we may need different strategies to make improvements. In general, for the first two types, which are relatively easy to identify because of the stylistic uniqueness and not so much ambiguity, we need to improve precision. For the third type, which can be identified by many regular NE taggers, but one must select the value associated with the target person, we have to improve the disambiguation ability. For the last type, involving unfamiliar NE categories, we have to develop a better recognition ability and improve recall first.

Also, we have to keep in mind that the precision for all the categories is very low, and good filtering techniques are needed to improve the accuracy.

Table8. Attribute Types

Description	Attributes (union recall)
There is a typical pattern which can be used to find the candidates	Phone (84.9), FAX (86.2), email (88.0), Web site (83.1),
Unfamiliar NE types, for which the candidates are relatively limited	Degree (65.4), Nationality (81.2)
Typical NE types, but disambiguation is needed to select only the attribute value	Date of birth (74.8), Birth place (69.9), othername (38.4), affiliation (44.6), school (60.2), mentor (18.7), relative (61.9)
Unfamiliar NE types, for which it is not easy to construct a list of possible vaues	Award (29.9), major (34.1), occupation (56.5)

Coverage Problem for high performing attributes (email)

It is a bit surprising that the recall of email is not very close to 100%. We found three major reasons for this. One is that some of the pattern matching can't detect some special symbols (e.g. minus sign). The second is that if a page contains a list of a large number of names with e-mail address in a table or other format, it is not easy to select only the e-mail of the target person. Systems rather gave up on detecting any e-mail address. The last problem is that some e-mail addresses are created by java scripts and it is not easy to detect them in the source page.

6.3 Coverage Problem for ambiguous attributes (affiliation)

The primary reason for missing coverage of affiliations is that many affiliation names for a wide variety people have a wide variety of types, including a very large number of minor ones. For example, a soccer player's affiliation, such as "Falkirk F.C" or "Scotland national football team" is quite difficult to detect. WePS covers a wide variety of people, including musicians, scientists, medical doctors and so on, and we are just not ready to recognize all those types of affiliations.

6.4 Coverage Problem for unfamiliar attributes (award)

The problem is similar to the one in 6.4. There are a very wide variety of award names, such as "Friend of Freedom Award", "Society of Professional Journalists First Lifetime Award", and we are not ready to recognize all such awards.

6.5 Accuracy Problem for high performing attributes (email)

In case of email, the systems rarely extract values of the wrong entity types. However, there are many email addresses mentioned in web pages, including email addresses of the web masters, contact persons, friends, persons who make comments, other people in a list, etc., so we need a smart way to filter those email addresses.

6.6 Accuracy Problem for ambiguous attributes (affiliation)

The type of affiliations is mostly organization. Many NE taggers are able to tag organization instances in texts. Although we can identify an organization, it is not easy to determine if it is the person's affiliation. First of all, the web pages have various structures, and unlike texts, such as newspaper and legal documents, the writing style depends on the page. Some of the information is expressed by tables and lists and sometimes position information is required to understand the relationship of different pieces of information. The systems often identify the wrong span of organization name because it appears within a web page, rather than well formed sentences. It may be necessary to develop an NE tagger for Web pages.

6.7 Accuracy Problem for unfamiliar attributes (award)

A good NE tagger is needed to identify names, but because *award* is usually an unfamiliar type of name for NE taggers, the systems are mostly not yet mature in tagging award names. Some systems use simple rules such as, first finding one of a set of capitalized words such as "Award" or "Prize", and then searching from right to left starting from the clue word for additional capitalized words. This strategy failed when the name includes non-capitalized words such as "of" or "for", or includes a year or ordinal number such as "5th". The clue words are sometime used in a different context such as "the 5th Annual Award Ceremony" etc. We need to develop a more accurate tagger for these types of names.

7. ACKNOWLEDGMENTS

We thank all the participants, annotators and other contributors to the project. In particular, Mr. Kosuke Takeuchi, Ms. Elizabeth Coogan Russell, Ms. Reiko Kawahara and Mr. Charles Shoopak for conducting very difficult annotations.

8. REFERENCES

[1] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. The SemEval-2007 WePS evaluation: Establishing a benchmark for the Web People Search Task. In SemEval, ACL, 2007.

[2] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. The WePS2 evaluation. In workshop on Web People Search, 2009.

[3] de Pablo-Sánchez, C. and Martínez P. Building a Graph of Names and Contextual Patterns for Named Entity Classification. To be published in ECIR 2009

[4] Satoshi Sekine. Extended Named Entities Ontologies with Attribute Information, In the proceedings of LREC 08.

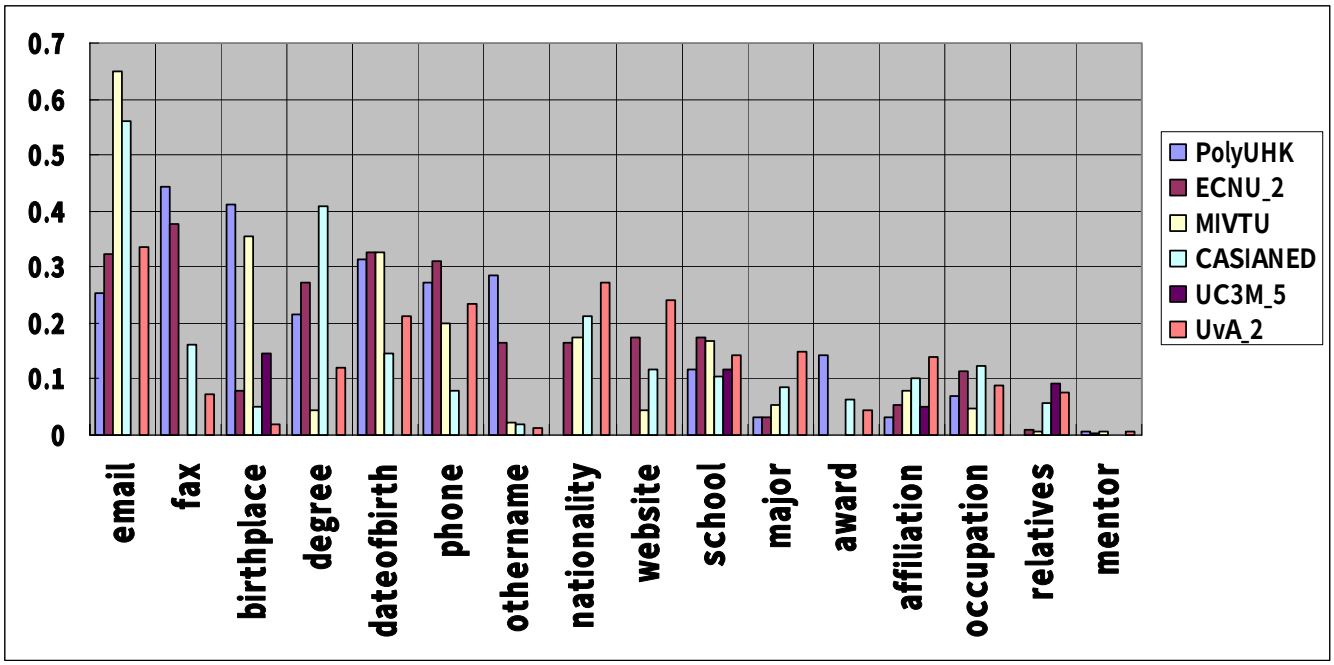

Figure 3. Results (F-measure) by Attribute Types

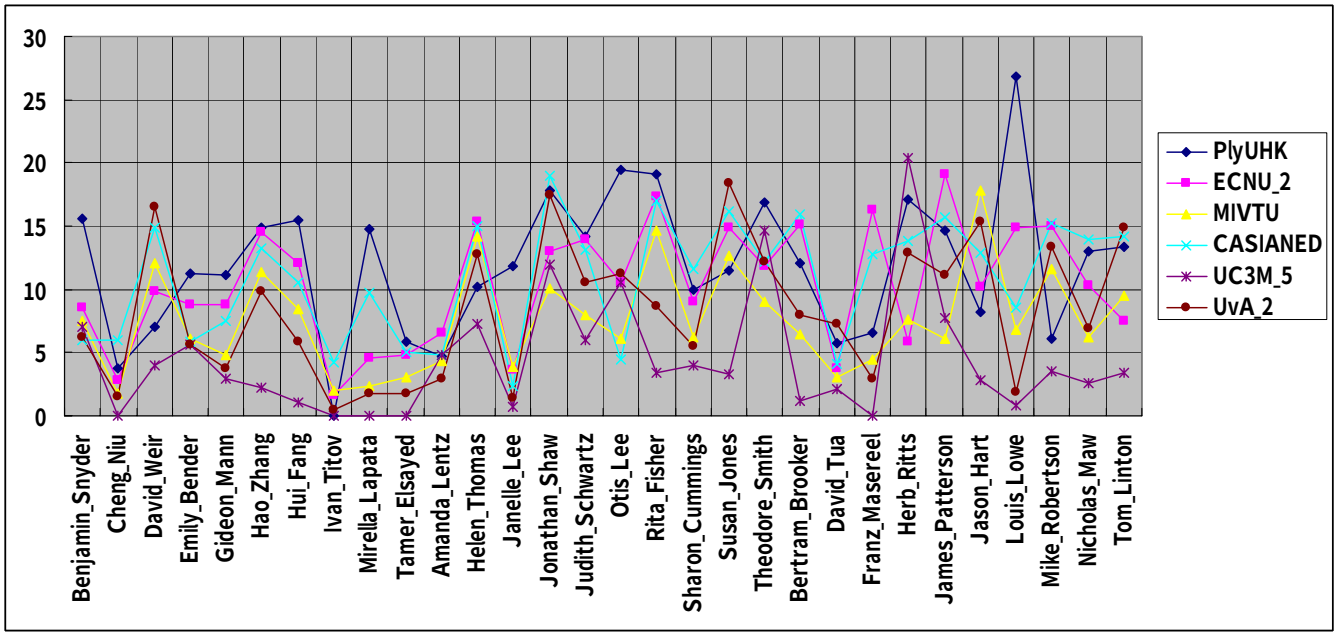

Figure 4. Results (F-measure) by People Names