Turk J Elec Eng & Comp Sci, Vol.19, No.1, 2011, © TÜBİTAK
doi:10.3906/elk-1002-10

Automatic knowledge extraction for filling in biography forms from Turkish texts

İlknur PEHLİVAN, Zeynep ORHAN
Department of Computer Engineering, Faculty of Engineering, Fatih University, Büyükçekmece, 34500, İstanbul-TURKEY
e-mail: zorhan@fatih.edu.tr

Abstract

This study presents a method for building an automatic knowledge extraction system for filling in biography forms from Turkish texts. Several biographies are analyzed in order to choose the set of biography categories to be studied. The fields of the biography form to be created are also defined based on this analysis. Information extraction techniques are used for implementation. A separate testing platform is designed to evaluate the accuracy of the extracted data. Results of the testing platform have shown this study to be a promising process to be further developed especially for creating forms in the Turkish language.

Key Words: *Information Extraction, Regular Expression, Biography Form Generation, Natural Language Processing*

1. Introduction

Information Extraction (IE) technology relies on the analysis of natural language in order to extract snippets of information [1]. The process inputs text and produces fixed-format, unambiguous output data. This data can be directly displayed, stored in a database such as a spreadsheet for later analysis, or used in Information Retrieval (IR) applications such as Internet search engines such as Google [2].

This study considers the ease and convenience IE systems can provide to users and studies IE techniques to create a way to summarize entities such as date of birth, name, job title, etc. from unstructured Turkish texts. Desired information along with their relevant entities is then extracted for filling in the fields, resulting in concise and precise forms of biographies. Figure 1 summarizes all the components of a typical IE system. These components and brief information about the data types, methods and applications used by the proposed biography generation system are explained in the following sections.

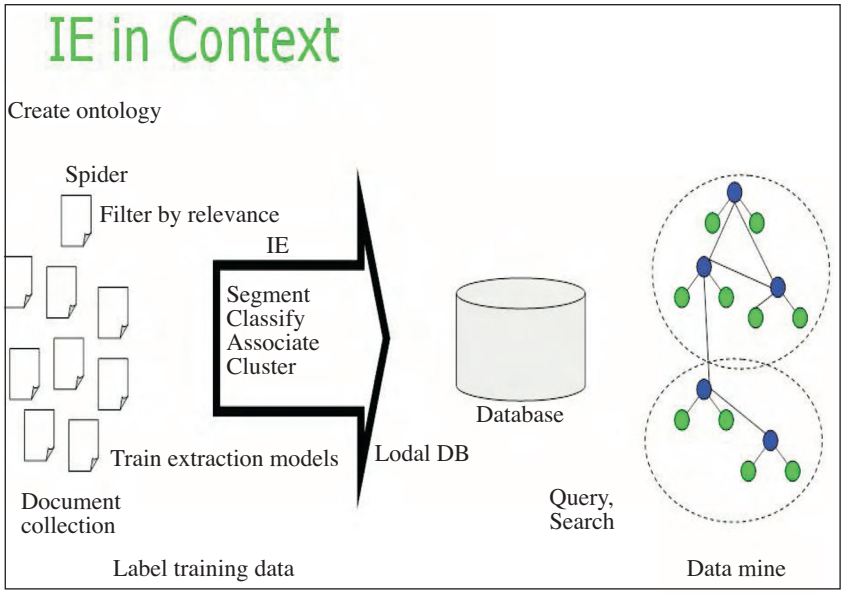

Figure 1. Information Extraction in context [3].

1.1. Data types

IE systems extract information from unstructured text and fill in the fields of a database with required data, such as name, title and organization [3]. Unstructured text, which is available in vast amounts and is used very frequently, is one of three types of data harvested by IE systems. Table 1 provides an overview of how these data types are encountered in everyday life.

Table 1. Data types of an IE system.

Data Types	Examples
Structured Text[4]	•Tabular data existing in databases •Spreadsheets •Schemas •Charts •Tables
Semi- Structured Text[5]	•Medical records •Equipment maintenance logs •Scientific data •XML codes available online
Unstructured Text[5]	•Web pages •Text documents •Office documents •....

1.2. System subtasks

An IE system first filters the information from documents, the extraction is trained by segmentation, classification and association; then the information is loaded into database fields; and finally, data mining is done to

obtain structured, useful information through some specific query search [6]. These phases are illustrated by an example in Table 2 and explained in detail as follows.

- Segmentation finds the starting and ending boundaries of the text snippets that will fill in a field of a form or database. This phase of the extraction is to delimit the beginning and end of the sentence or a segment to be extracted. School name or graduation information must be extracted, and segmentation must find the first and last words of the title, avoiding inclusion of extra words or chop off too many words.

- Classification determines which field is the correct destination for each text segment. For example, "*İstanbul'da doğdu*" ("S/he was born in Istanbul") belongs to the "placeofbirth" field; "*20 Kasım 1945*" ("November 25, 1945") belongs in the "dateofbirth" field; and "*Osmanlı Bankası müdürü olarak...*" ("as director of the Ottoman Bank...") belongs in the "experience" field.

- Association determines which fields belong together in the same record. Sometimes the same information content is expressed in different words. To associate this kind of information and to avoid redundancy, the fields must be examined for grouping similar contents. A standard format is needed for the data to be reliably compared. The information locked in natural language must first be transformed into standard, structured and normalized form.[7]

Table 2. Information extraction subtasks [3].

October 14, 2002, 4:00 a.m. PT For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation. Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels—the coveted code behind the Windows operating system—to select customers. "We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access. "Richard Stallman, founder of the Free Software Foundation, countered saying...		
Segmentation	Classification	Association
Microsoft Corporation CEO Bill Gates Microsoft Gates Microsoft Bill Veghte Microsoft VP Richard Stallman founder Free Software Foundation	Microsoft Corporation CEO BILL GATES Microsoft GATES Microsoft BILL VEGHTE Microsoft VP RICHARD STALLMAN founder Free Software Foundation.	Microsoft Corporation CEO BILL GATES Microsoft GATES Microsoft BILL VEGHTE Microsoft VP RICHARD STALLMAN founder Free Software Foundation

1.3. Related work

Extracting information from texts or web pages to generate various reports is becoming the focus of many research interests. There is not much work found in similar subject during the literature survey of this study.

However, based on the research performed, it could be said that this is the first and, at present, only study conducted for application to the Turkish language. The following is shown to name the previous studies;

Artequakt is an example, which inspired this study with its idea of implementing a system that searches the Web and extracts knowledge about artists, based on an ontology describing that domain, and stores this knowledge in a KB to be used for automatically producing personalized biographies of artists [1].

RAPOSA is another successful study, an open-domain automatic question-answering system for Portuguese; it is intended to be a key component of a larger information extraction framework that uses question-answering technology as the basis for automatic generation of biographies [4]. This question-answering technology has been an inspiration for this study when generating rules of regular expressions.

GATE is one of the oldest (since 1995) and most stable project; it has been used for many IE projects in many languages and problem domain, and has competed in the MUC and ACE evaluations [8]. GATE has a built-in IE component set called ANNIE, which makes it compatible and desirable at the same time.

2. Implementation

Six biography categories have been chosen to be analyzed in this study: *Presidents, Politicians, Authors, Poets, Actors, and Singers*, which are found to be the most frequently referred biography types by users. Analyzing these biographies led to the observation that the most important emphasis is put on the following six particular fields:

- *Date of Birth*
- *Date of Death*
- *Education*
- *Experience*
- *Contributions*
- *Rewards.*

The same facts in biographies can be presented in various ways, as it can be easily observed from the aforementioned examples, and hence is a serious problem for resolving conflicts between extracting the correct set of information and filling in the corresponding fields of the extracted set. The following example further illustrates this by a set of data extracted from the biographies analyzed. The birth date *November 20, 1954* may be presented as *20 Kasım 1954* or *20/11/1954* or *Kasım'ın 20 sinde 1954 yılında* and they all carry the same fact of being a birth date. It is desired that a search to be able to detect this kind of variation. Unfortunately, simple string comparisons may not be capable of achieving this task; conversion of data to the standard representation is required. This representation is done by writing regular expressions manually for each field of the form. Figure 2 shows the overall procedure followed for the information extraction process:

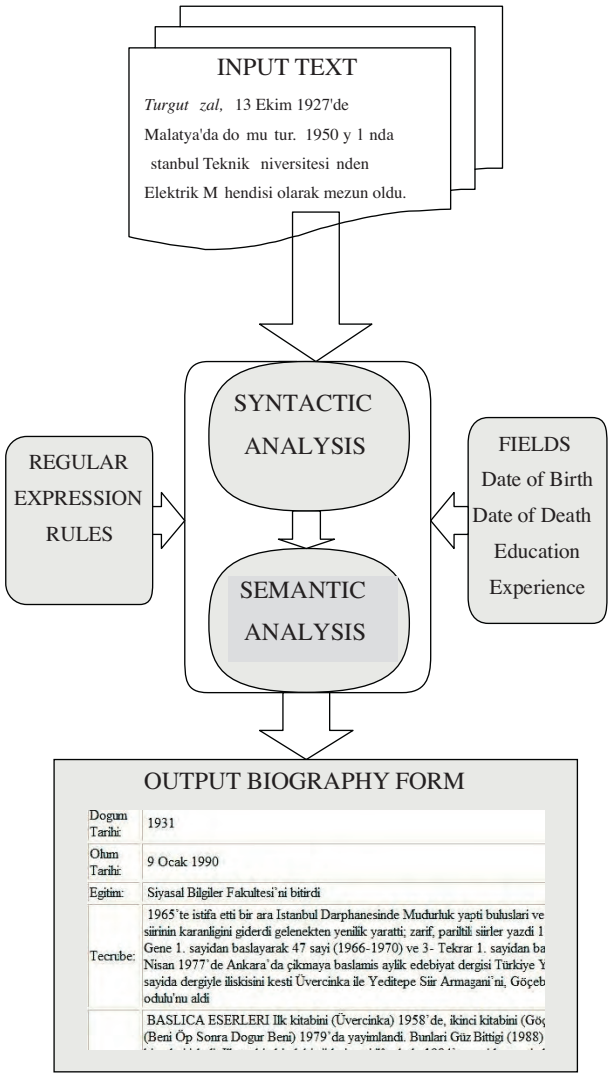

Figure 2. Diagram of Information Extraction process for automatic biography form generation.

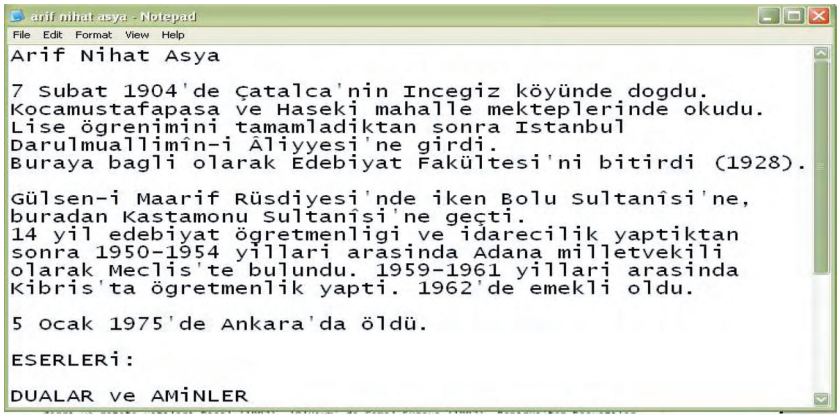

Figure 3. Screenshot of a Turkish text used as input to generate biography form.

2.1. Regular expressions

Regular expressions (RE) provide concise and flexible means for identifying strings in a text of interest, such as particular characters, words, or patterns of characters. REs are written in a formal language that can be interpreted by a RE processor (engine). This processor is a program that either serves as a parser generator or examines text and identifies parts that match the provided specification [9].

Almost all programming languages provide many built-in string functions. They are very useful to solve simple to moderately-complex programming tasks. If the string to be searched is something like "*X university computer club"* then it is still extractable by the string function. If part of the data set to be fetched is unknown to the user or not limited to a fix set of data set then string functions cannot be used. For example; it is difficult to fetch all the set of strings which match the *"X university... club"* pattern by standard string operations. Searching for the terms "*X*" and *"club"* separately, produces many irrelevant results.

Where there is some sort of complication or an arbitrary case such as the one presented above, REs come in very handy for an accurate result. They are usually used to give a concise description of a set, without having to list all elements. For example, the set containing the three strings "ö*zcan*," "ö*zhan*" and "ö*zkan*"[1] can be described by the pattern **öz[chk]an.** Alternatively, it is said that the pattern matches each of the three strings or even the words as in the previous example. The RE "**X university ([a-zA-Z]+) club**" will list all the clubs of X university.

2.2. Regular expression operations

The most frequently used RE operations are as follows [10]:

- **Boolean "or":** A vertical bar separates alternatives. For example, **kerim|kerem** can match "*kerim*" or "*kerem.*"
- **Grouping:** Parentheses are used to define the scope and precedence of the operators (among other uses). For example, **kerim|kerem** and **ker(i|e)m** are equivalent patterns which both describe the set of "*kerim*" and "*kerem.*"
- **Quantification:** A quantifier after a token (such as a character) or group specifies how often that preceding element is allowed to occur. The most common quantifiers used in this study are presented in Table 3.

Table 3. Table of regular expression operators.

?	The question mark indicates there is zero or one of the preceding element. For example, **ab?c** matches both *"ac"* and *"abc"* but not *"abbc"*.
*	The asterisk indicates there is zero or more of the preceding element. For example, **ab*c** matches *"ac", "abc", "abbc", "abbbc"*, and so on.
+	The plus sign indicates that there is one or more of the preceding element. For example, **ab+c** matches *"abc", "abbc", "abbbc"*, and so on, but not *"ac"*.

[1] Regular expressions are given as bold and Strings are given as italic

2.3. Rule representation by using regular expressions

There are six rules written for six fields by using regular expressions. Below, lines show a sample of lists made from the segments taken from over 50 Turkish texts.

2.3.1. Field 1: DateofBirth Rule

List of different sentences expressing fact of birth information:

<Tarih ifadesi>'de hal eki doğ <dili | mişli geçmiş zaman>

<Tarih ifadesi>'de hal eki dünyaya gel <dili | mişli geçmiş zaman>

<Tarih ifadesi>'de hal eki dünyaya gözlerini aç <dili | mişli geçmiş zaman>

<Tarih ifadesi> tarihinde doğ <dili | mişli geçmiş zaman>

<Tarih ifadesi> tarihinde dünyaya gel <dili | mişli geçmiş zaman>

<Tarih ifadesi> tarihinde dünyaya gözlerini aç <dili | mişli geçmiş zaman>

<Tarih ifadesi> yılında doğ <dili | mişli geçmiş zaman>

<Tarih ifadesi> yılında dünyaya gel <dili | mişli geçmiş zaman>

<Tarih ifadesi> yılında dünyaya gözlerini aç <dili | mişli geçmiş zaman>

<Tarih ifadesi>'de hal eki < şehir/ülke adı >'de hal eki doğan....

Summarized general expression of the rule:

DateofBirthRule

<Tarih ifadesi> ((‘| tarihin|yılın|senesin) DE HALEKİ)) kelime[2] (doğ | (dünyaya|hayata (gel | gözlerini aç)) (DİLİ | MIŞLI GEÇMIŞ ZAMAN EKI | (an | en))

DateofBirth rule written in regular expressions:

(((0?[1-9])|([12][0-9])|(3[01]))?(/|\\.|\\−|.)*(0?[1-9]|1[0-2] |

(Ocak|Şubat|Mart|Nisan|Mayıs|Haziran|Temmuz|Ağustos|Eylül|Ekim|Kasım|Aralık)*)?(/|\\.|\\−|.)* ([12][0-9][0-9][0-9]))?((\'d(a|e))*|tarihin|yılın|senesin)(‘d[ea])*(doğdu| dünyaya|hayata (geldi|gözlerini açti)|\\s

As complicated and confusing as it may look, the above expression is very effective, powerful and precise way of extracting desired words in a particular pattern from an unstructured text. Detailed explanation of the above rule is given below:

<Tarih ifadesi> (Date pattern) is to fetch any kind of date information which can either be 25/11/1924 or 25-11-1924 or 25.11.1924 or even 25 Ocak 1924.

Day (dd) part of the date is defined:

“(0?[1-9])” // defines days written as 1, 2, .., 9, 01, 02, ..., 09.

“([12][0-9])” // defines days written as 10, 11, ..., 29

“(3[01])” // defines days written as 30, 31

These three definitions are combined by | OR to set the pattern for the days of a year.

Pattern of Day: “((0?[1-9])|([12][0-9])|(3[01]))”

Month of a year is defined as the following:

“(0?[1-9])” //defines months written as 1, 2, .., 9, 01, 02, .., 09.

[2]Maybe any word or set of words such as “4 kardesin en buyugu olarak.”

"(1[0-2])" //defines months written as 10, 11, 12...

These two definitions are combined by | OR to set the pattern for the months of a year. Since names of months are also to be extracted as in the form of "*12 Ocak 1980*" list of month names are also included in the pattern of month.

(Ocak|Şubat|Mart|Nisan|Mayıs|Haziran|Temmuz|Ağustos|Eylül|Ekim|Kasım|Aralık)
Pattern of Month: "(0?[1-9])|(1[0-2])|
(Ocak|Şubat|Mart|Nisan|Mayıs|Haziran|Temmuz|Ağustos|Eylül|Ekim|Kasım|Aralık)"

Year pattern is given below:

"([12][0-9][0-9][0-9]) this is to contain all the years between 1000–2999

Combination of these three patterns gives the complete date format as:

"((0?[1-9])|([12][0-9])|(3[01]))(/)(0?[1-9]|1[0-2]|
(Ocak|Şubat|Mart|Nisan|Mayıs|Haziran|Temmuz|Ağustos
|Eylül|Ekim|Kasım|Aralık)) (/)([12][0-9][0-9][0-9]))"

Complete Date pattern:

(((0?[1-9])|([12][0-9])|(3[01]))?(/|\\.|\\−|.)*(0?[1-9]|1[0-2]
|(Ocak|Şubat|Mart|Nisan|Mayıs|Haziran|Temmuz|Ağustos|Eylül
|Ekim|Kasım|Aralık)*)?(/|\\.|\\−|)*([12][0-9][0-9][0-9])))

Complete regular expression for the field 1, DateofBirth rule:

(((0?[1-9])|([12][0-9])|(3[01]))?(/|\\.|\\−|.)*(0?[1-9]|1[0-2]|
(Ocak|Şubat|Mart|Nisan|Mayıs|Haziran|Temmuz
|Ağustos|Eylül|Ekim|Kasım|Aralık)*)?(/|\\.|\\−|.)*
([12][0-9][0-9][0-9]))?((\'d(a|e))*|tarihinde|yılında|senesinde)*
(doğdu|dünyaya geldi|gözlerini actı)|\\s)*))

Below are some examples to what the above date pattern is capable of fetching from text:

20/12/1940 |yılında doğdu.
20.12.1940 tarihinde dünyaya geldi
20-12-1940'da doğdu
20 Aralik 1940'da dünyaya gözlerini actı.

2.4. Regular expression applications

Regular Expressions (REs) can be applied to various domains successfully, since they have a very strong theoretical ground and simple usage. Ambiguity cases are perfect job for REs [9], all of the URLs (web addresses) could be searched by short description of the URL pattern, e-mail address verifications are done by fewer lines of code, different data types can be converted via REs, data carried by XML lines can be converted to arrays in php [11]

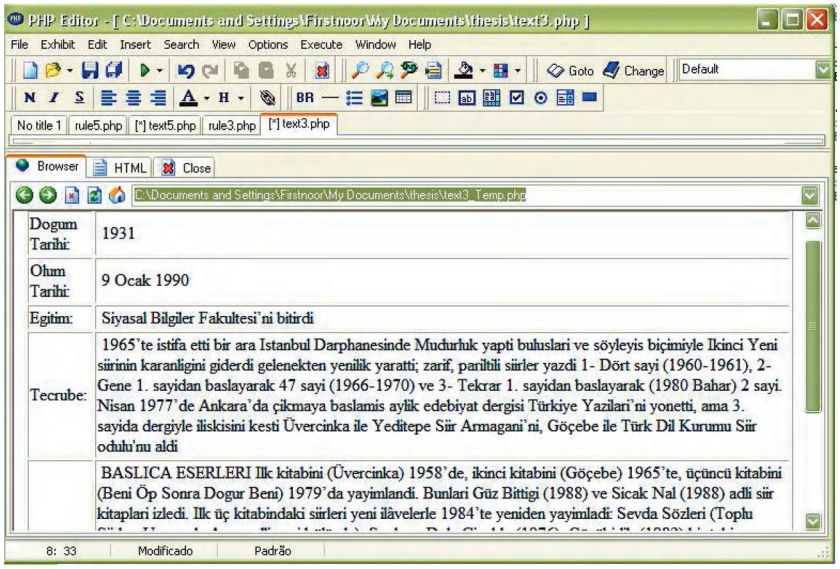

Figure 4. Generated biography form.

3. Results and comparisons

This section will provide detailed and figurative explanation of the biography forms obtained, a testing platform designed to measure the correctness of the generated biography forms and comparison of the forms and original data extracted manually from over 50 documents for six people in six different categories.

The input to the system is the biography as a text file and the output is a form that has the required fields filled in with the data extracted from the related text as demonstrated in Figures 3 and 4 respectively. Followings are the manually formed tables of *president category,* there are over 30 tables created similar to these ones for six categories as mentioned previously.

Table 4. Original data for date of birth and death fields (President category).

Original Data		
President	Date of birth	Date of Death
Cemal Gürsel	1895	14 Eylül 1966
Fahri Korutürk	1903	12 Ekim 1987
Süleyman Demirel	1924	—
Cevdet Sunay	1899	22 Mayıs 1982
Turgut Özal	1927	17 Nisan 1993
Celal Bayar	16 Mayıs 1883	22 Ağustos 1986

A separate testing platform is specially designed to evaluate the accuracy and quality of the results obtained. Several tables are created and placed as excel files. Half of these tables are filled with the fields of biography forms; the other half contains the data manually extracted from Turkish texts. The tables similar to Table 4–Table 7 are used as input to the testing platform; these tables are shown to compare the original and matched data for the *president* category. They are also used as input to the testing platform as two tables per spreadsheet, content of each cell of the tables is compared word by word.

Table 5. Matched data for date of birth and date of death fields (President category).

Matched Data		
President	Date of birth	Date of Death
Cemal Gürsel	1895	14 Eylül 1966
Fahri Korutürk	1903	12 Ekim 1987
Süleyman Demirel	1924	—
Cevdet Sunay	1899	22 Mayıs 1982
Turgut Özal	1927	17 Nisan 1993
Celal Bayar	16 Mayıs 1883	22 Ağustos 1986

Table 6. Original data for education field (President Category).

Original Data	
President	Education
Cemal Gürsel	İlk öğrenimini Ordu'da bitirdi. Daha sonra egitimini Erzincan ve Istanbul'da askerî öğrenci olarak tamamladı. 1929 yılında Harp Akademisi'ni bitirdi.
Fahri Korutürk	Heybeliada Bahriye Mektebi'nde tahsil gördü. 1923 yılında Gv.Mühendis (Teğmen) rütbesiyle mezun olmuştur. 1931 yılında girdiği Deniz Harp Akademisini 1933 yılında bitirdi ve Kurmay Subay oldu.
Süleyman Demirel	İlk öğrenimini doğduğu köyde, ortaokul ve liseyi Isparta ve Afyon'da bitirdi. Şubat 1949'da Istanbul Teknik Üniversitesi İnşaat Fakultesi'nden mezun oldu. Amerika Birlesik Devletleri'nde barajlar, sulama ve elektrifikasyon konularinda ihtisas yaptı.
Cevdet Sunay	İlk ve orta öğrenimini Erzurum, Kerkük, Edirne ve Kuleli Askerî Lisesi'nde yaptı.1927 yılında Harp Okulu öğrenimini tamamladı. 1930 yılında Harp Akademisi'ni bitirdi.
Turgut Özal	Istanbul Teknik Üniversitesi'nden Elektrik Mühendisi olarak mezun oldu. 1952 yılında A.B.D'ne giderek ekonomi tahsili gördü.
Celal Bayar	İlk ve orta öğrenimini babası Abdullah Fehmi Efendi'nin yanında tamamladı.Bir ara Harir Darutariri okulunda okudu.

Table 7. Matched data for education field (President category).

Matched Data	
President	Education
Cemal Gürsel	İlk öğrenimini Ordu ilinde yaptı. Daha sonar öğrenimini Erzincan ve İstanbul'da askerî öğrenci olarak sürdürdü. 1929 yılında Harp Akademisi'ni bitirdi.
Fahri Korutürk	Heybeliada Bahriye Mektebi'ndeki tahsilini takiben 1923 yılında Gv.Mühendis (Teğmen) rütbesiyle mezun olmuştur. 1931 yılında girdiği Deniz Harp Akademisini 1933 yılında bitirdi
Süleyman Demirel	İlk öğrenimini doğduğu köyde, ortaokul ve liseyi Isparta ve Afyon'da bitirdi. Şubat 1949'da İstanbul Teknik Üniversitesi İnşaat Fakultesi'nden mezun oldu.
Cevdet Sunay	İlk ve orta öğrenimini Erzurum, Kerkük, Edirne ve Kuleli Askerî Lisesi'nde yaptı. 1930 yılında Harp Akademisi'ni bitirdi. 1930 yılında Harp Akademisi'ni bitirdi.
Turgut Özal	1950 yılında İstanbul Teknik Üniversitesi'nden Elektrik Mühendisi olarak mezun oldu. 1952 yılında ekonomi tahsili gördü.
Celal Bayar	İlk ve orta öğrenimini babası Abdullah Fehmi Efendi'nin yanında yaptı. Harir Darutariri okulunda okudu.

Then comparison tables are generated showing the total number of words taken from the fields of the generated biography forms over total number of words manually taken from the original texts as shown in Table 8.

The testing platform takes two tables per sheet for each field of a biography category as it is mentioned previously. The content of each cell of these tables are compared word by word. Table 8 represents the total number of words counted from the field, containing original data, over number of words taken from the fields of generated biography forms, which are matching with the original data. Formulating this expression that is Recall (R) or coverage in the literature will give

$$\frac{\text{Number of words matching the original data}}{\text{Number ofcwords in the original data}}.$$

The information demonstrated in each cell of the comparison table (Table 8) can be explained more precisely as follows:

- 1/1 in the *Date of Birth* field means only *year* information is present and it is matched.
- 3/3 means Date of Birth field contains all day, month and year information and they are all matched.
- 193/256 for the Experience field shows that a total of 193 words are matched out of 256 words.
- Finally, as it could be seen contributions and rewards fields are shown as "N/A/0"it means the field is null and nothing is matched from the text or in other words there was no such information to be found in the given text.

Table 8. Comparison table for President category.

Comparison of original & matched data (word/word)						
President	Date of Birth	Date of Death	Education	Experience	Contributions	Rewards
Cemal Gürsel	1/1	3/3	19/19	83/105	N/A/0	N/A/0
Fahri Korutürk	1/1	3/3	20/24	77/118	N/A/0	N/A/0
Süleyman Demirel	1/1	N/A/0	30/30	173/205	N/A/0	N/A/0
Cevdet Sunay	1/1	3/3	20/22	76/94	N/A/0	N/A/0
Turgut Özal	1/1	3/3	16/16	152/193	N/A/0	N/A/0
Celal Bayar	3/3	3/3	15/15	193/256	N/A/0	N/A/0

It is observed from the above tables that, matching *Date of Birth* and *Date of Death* fields are generating the most accurate results. *Contributions* and *Rewards* fields are producing a better quality, because these fields do not necessarily exist for all the entries. The *Education* and *Experience* fields are the most challenging ones to match, since they bear among the most populated content, and is more difficult to obtain 100% match.

The Precision/Recall values of extracted relations from around 50 documents for 5 artists in the Artequakt Project are given in Table 9. The results of this project are the only ones available that have the most relevance to this study. The recall values of extracted relations for the *President* category of this study are provided in Table 10 for comparison.

In the Artequakt Project the precision is given importance and recall values are sacrificed for the sake of precision. This is to reinforce the concept that inaccurately extracted knowledge may reduce the quality of the

system's output. For this reason, their extraction rules were designed to be of low risk levels to ensure higher extraction precision. They also emphasized that the preference of precision versus recall could be dependent on the relation in question [1].

Table 9. Precision/Recall of extracted relations from around 50 documents for 5 artists in Artequakt Project [1].

Relation Artist (P/R)	Rembrandt (P/R)	Renoir (P/R)	Cassatt (P/R)	Goya (P/R)	Courbet (P/R)	Average
Date of birth	75/43	100/50	100/67	80/40	100/100	91/60
Place of birth	100/63	100/14	100/50	100/40	100/63	100/46
Date of death	100/63	100/67	100/50	N/A	100/50	100/46
Place of death	100/100	100/43	N/A/0	100/20	100/33	100/49
Place of work	100/50	67/33	33/100	N/A/0	0/0	40/37
Place of study	100/20	100/14	100/75	100/20	100/29	60/32
Date of marriage	100/50	100/33	N/A1	100/100	N/A/0	60/46
Name of spouse	100/38	N/A/0	N/A	N/A/0	N/A/0	100/10
Parent profession	100/57	50/67	0/0	67/100	100/100	63/65
Inspired by	100/43	50/60	0/0	100/17	100/33	83/31
Averages	98/53	85/38	61/43	92/34	88/41	85/42

Table 10. Recall values of extracted relations for President category. (Precision=100%).

President	Cemal Gürsel	Fahri Korutürk	Süleyman Demirel	Cevdet Sunay	Turgut Özal	Celal Bayar	Average
Date of Birth	100	100	100	100	100	100	100
Date of Death	100	100	100	100	100	100	100
Education	100	83	100	91	100	100	96
Experience	79	65	84	81	79	75	77
Contributions	N/A/0	N/A/0	N/A/0	N/A/0	N/A/0	N/A/0	N/A
Rewards	N/A/0	N/A/0	N/A/0	N/A/0	N/A/0	N/A/0	N/A
Average	95	87	96	93	95	94	

In our study, precision values are 100%. The average of the recall values of the date expressions and education are obtained as 100% and 96%, respectively. These values are considerably higher or competitive with the Artequakt Project. The other fields also achieve very good performance in terms of precision and recall, but it is not possible to find one-to-one correspondence with the Artequakt Project. Other categories are not shown here, since their results are pretty much similar to the ones depicted in the representative results of the Table 10.

4. Conclusion

The idea on building an automatic knowledge extraction for filling in biography forms from Turkish texts is presented in this study.

The system is implemented by using IE techniques or so-called Text Mining techniques. Fields of biography forms are filled in by using regular expressions. The rules are tailored according to the structure of desired data blocks. Data is then extracted for each field by running these RE rules on Turkish texts.

A testing platform is designed to evaluate the results. Excel tables are used as input to the testing platform. And the results are shown as tables of comparison. Evaluation results are expressed as tabular data by displaying the number of the words matched over number of words counted from original data.

The easier fields to obtain are *Date of Birth* and *Date of Death*, and the result of these fields are perfect or almost-perfect matches. Some of the fields, such as *Experience* and *Contributions,* present various behaviors and are very difficult to match compared to the others; generalizing the patterns of these fields is a difficult task. Remaining fields, namely, *education* and *rewards* are rather less complicated, because these fields carry fewer words, which is an advantage for comparison.

This study differs from existing systems in that it aims to extract specific facts and populate a knowledge base with these facts to be used in the generation of personalized biographies. The system provides promising results and can be easily used in many applications of the same type. The system is scalable and flexible enough to be used for various texts, the only modification required for including them is the modifications of REs.

References

[1] H. Alani, S. Kim, D. E. Millard, M. J. Weal, P. H. Lewis, W. Hall, N. R. Shadbolt, "Automatic Extraction of Knowledge from Web Documents", In: 2nd International Semantic Web Conference - Workshop on Human Language Technology for the Semantic Web and Web Services, October 20-23, Sanibel Island, Florida, USA, 2003.

[2] C. Hargood, D. Millard, M. Weal, "A Thematic Approach to Emerging Narrative Structure". In: Web Science Workshop at Hypertext, Pittsburg, USA, 2008

[3] M. Hearst, "Applied Natural Language Processing Lecture notes", Slides adopted by W. Cohen, 15 November 2006, http://www.sims.berkeley.edu/∼hearst

[4] L. Sarmento, "Hunting Answers with RAPOSA (FOX)", Working Notes of the Cross-Language Evaluation Forum Workshop CLEF, Alicant, Spain, 20-22 September, 2006

[5] S. Soderland, "Learning Information Extraction Rules for semi structured and free text", Special issue on natural language learning on Machine Learning, February 1999, Volume 34, Issue 1-3, Pages: 233 – 272, 1999,

[6] W. Cohen, "Fast effective rule induction", In Proceedings of the Twelfth International Conference on Machine Learning (ICML-95), pages 115–123, San Francisco, CA, 1995.

[7] A. McCallum, "Information Extraction: Distilling Structured Data from Unstructured Text", Social Computing , Q focus: social computing, Queue, Volume 3, Issue 9, Pages: 48 – 57, 2005

[8] H. Cunningham, "Information Extraction, Automatic", In Brown, K. (ed.), Encyclopedia of Language and Linguistics, vol. 1-14, p.665-677, 2nd Edition, Elsevier Science Publishers, 2005

[9] L. Karttunen, J-P. Chanod, G. Grefenstette, A. Schiller, "Regular expressions for language engineering", Natural Language Engineering,1996

[10] S. Levithan, J. Goyvaerts, Regular Expressions Cookbook, O'Reilly Media , 2003-2009

[11] D. Jurasfsky, and J.H. Martin, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall, New Jersey, 2000