Event Extraction from Classical Arabic Texts

Razieh Baradaran[1] and Behrouz Minaei-Bidgoli[2]
[1]Department of Information Technology, University of Qom, Iran
[2]Department of Computer Engineering, Iran University of Science and Technology, Iran

Abstract: *Event extraction is one of the most useful and challenging Information Extraction (IE) tasks that can be used in many natural language processing applications in particular semantic search systems. Most of the developed systems in this field extract events from English texts; therefore, in many other languages in particular Arabic there is a need for research in this area. In this paper, we develop a system for extracting person related events and their participants from classical Arabic texts with complex linguistic structure. The first and most effective step to extract event is the correct diagnosis of the event mention and determining sentences which describe events. Implementation and comparing performance and the use of various methods can help researchers to choose appropriate method for event extraction based on their conditions and limitations. In this research, we have implemented three methods including knowledge oriented method (based on a set of keywords and rules), data-oriented method (based on Support Vector Machine (SVM)) and semantic oriented method (based on lexical chain) to automatically classify sentences as on-event or off eventones. The results indicate that knowledge oriented and machine learning methods have high precision and recall in event extraction process. The semantic oriented method with acceptable precision minimizes the linguistic knowledge requirements of knowledge oriented method and preprocessing requirements of data oriented method; and also improves automatic event extraction process from the raw text. Next step is developing a modular rule based approach for extracting event arguments such as time, place and other participants involved in independent subtasks.*

Keywords: *Event extraction, SVM, lexical chain, rule based method, classical arabic texts.*

Received February 18, 2013; accepted September 19, 2013; published September 4, 2014

1. Introduction

Nowadays, large data volumes which can be very easily accessed are available. But searching and finding appropriate and required information inside these sources have become a challenging problem; therefore, Information Extraction (IE) was created and developed to extract pre-specified information from the raw text in order to organize it in a structured format. Event extraction is one of the most important and difficult tasks which identifies events and related information by the raw text.

First some event extraction basic concepts are explained and then event extraction process and use are introduced.

According to Automatic Content Extraction (ACE) standard the following concepts are defined:

- Event: Something that happens in the specific time and place like occurrence of a particular crime or incident.
- Event Trigger: The main word which most clearly expresses an event occurrence.
- Event Argument: The mentions which are involved in an event (participants).
- Event Mention: A phrase or sentence within which an event is described, including triggers and arguments.

One of the most important applications of event extraction system is in the information search and question answering systems. Due to the large volume of documents, certain systems to automatically detect and extract relevant information are needed. In fact, automatic event extraction from the text is an essential need for new search systems, therefore events and relations are considered to be document metadata which facilitate search in general and semantic search in particular [5].

Event extraction can be used in text summarization process. In this case, extracted event with related description are the basis of summarized text. This technique can be applied in various domains such as extracting news events from the related documents and increasing the efficiency of news systems, so that the news messages can be selected more carefully and based on user preferences. In addition, event extraction is used widely in the medical domain [5] and extracted medical and biological events are used in many researches. Event extraction is also used in other domains such as e-commerce applications [14] and sport events detection [1].

Event extraction generally consists of two stages. In the first stage, sentences that describe specified event are marked and in the second stage, event arguments such as time, date, place and other participants involved in events are extracted.

1.1. Related Works

Although, event extraction is relatively new, valuable research has been done in this area. Automatic event extraction has started from Topic Detection and Tracking (TDT) project in [23] which is done by the National Institute of Standards and Technology (NIST). This project develops technologies to extract events from news stream and track the progression of these events over time.

In this project, a set of pre-defined templates are used for specified event information. IE process has been done by mapping different rules and grammars.

Then, automatic event extraction was followed by ACE program in 1999 [3]. The objective of this program was to develop technologies to process and extract desired information from the natural language documents. This information includes entities and relationships which were gradually incorporated into the ACE program; and finally in 2004 eVent Detection and Recognition (VDR) task were also added.

It can be said ACE program is a way to standardize IE tasks, which is used to evaluate other systems.

In addition to the two mentioned systems, more researches have been conducted in this area, which use different methods for event extraction. These methods are different in linguistic level, the number of ontologies, required lexical resources, supervision level of human and domain specificity.

According to Hogenboom *et al.* [12] event extraction methods used in these researches, can be classified as follows:

- Data Oriented Event Extraction: Data oriented event extraction approaches rely on quantitative methods in order to discover relations. These approaches require large text corpora in order to develop models that approximate linguistic phenomena. The main problem with these methods is that they do not deal with meaning in the discovering process. They also, require large volumes of data. However, since these methods are not based on knowledge, neither linguistic resource, nor domain-expert knowledge is required; hence, it can be a proper way for people who are unfamiliar with linguistic information.
- Knowledge Oriented Event Extraction: Unlike data oriented method, knowledge oriented text mining, relies on patterns and rules which represent expert terminology and linguistic knowledge. Information is extracted from the text using predefined or discovered linguistic patterns. This approach reduces the problems of statistical methods which result from the lack of attention to the meaning of the text. It also, requires much less data in comparison with data oriented methods. But it needs lexical knowledge and possibly prior domain knowledge to define patterns which are able to retrieve desired and proper information. Other disadvantages are related to maintaining and improving patterns; and that these patterns are usually domain specific and may not cover all situations.
- Hybrid Event Extraction: Despite the advantages of both data oriented and knowledge oriented approaches to event extraction, usually a combination method is applied to overcome the disadvantages of both approaches and to achieve the best result. In fact, hybrid approach is a compromise between two previous approaches.

Related articles and researches categorized based on the three mentioned approaches are shown in Table 1.

Table 1. An overview of event extraction researches and their approaches.

Event Extraction Approach	**Research**	**Year**	**Method**	**Event**
Data Oriented	Ahn [4]	2006	Nearest Neighbor and Maximum Entropy Classifiers	ACE
	Naughton *et al.* [17]	2008	SVM and Linguistic Modeling	ACE
Knowledge Oriented	Aone and Ramos-Santacruz [6]	2000	Lexicon-Syntactic Patterns	Different Domain
	Vargas-Vera and Celjuska [22]	2004	Semantic-Syntactic Patterns	Kmi News Event
	Xu *et al.* [25]	2006	Using Seeds and Pattern Learning Bysystemminipar	News Events of Nobel Reward Domain
	Abuleil [2]	2007	Pattern Matching in Arabic News Document	Different Domain
Hybrid	Piskorski *et al.* [18]	2007	Clustering-Pattern Matching	Security Related in News Document
	sangeetha *et al.* [20]	2010	Using Lexical Chain to Trigger Detection and Pattern Matching to Argument Detection	Wikipedia Document Local Event
	sangeetha *et al.* [21]	2010	Trigger Detection using Bayesian Theory and Argument Detection using Ontosem System and Text Meaning Representation	ACE

Classifying sentences describing an event is an important step in event extraction process and influences the accuracy of the process.

According to different IE approaches and methods, and each specific application, a researcher is faced with various options. Implementation of these methods and comparing their performance and use can provide the researcher with a better view and help him select a method based on expected possibilities and use.

1.2. Contributions

Most event extraction systems support English and European language texts. For many other languages like Arabic and South Asian languages, a research need is felt. In addition, Arabic language is an ancient language with many historical documents which provide valuable information for researchers. So, systems to automatically extract information are required.

Our event extraction system is the first one in historical Arabic texts. Due to specific linguistic characteristics and more complex linguistic rules of Arabic literature in comparison with others, event extraction in this language is more complex. In addition, a historical Arabic text has a relatively different structure from a modern Arabic text and that increases the complexity.

On the other hand, we use raw texts with unstructured format as input dataset which have different linguistic structures. In other word, our system can extract events from any historical Arabic text.

In this research we develop relatively complete set of linguistic patterns to extract person related events in particular die event in historical Arabic text with the help of linguistic expert team. So, event extraction can be done with high accuracy (above 90% precision for die event) by this system.

This research consists of two main stages: The first step is classifying sentences as on-event or off-event with three different methods and comparing their performance and use. The first method is a rule based one which uses a set of trigger keywords and linguistic rules regarding each of them in order to determine sentences containing events. The second method is a machine learning classifier for learning on-event sentences using linguistic features of words or sentences and evaluating their accuracy using test. documents. As the last method, a semantic oriented approach is used to determine on-event sentences. This method applies a lexical chain based on training text, to classify test sentences.

1.3. Organization

In the next stage we implement our event extraction system with a modular approach. So, event information is extracted independently and does not affect the other extraction process. The outline of the paper is as follows: The next section explains data and materials, section 3 describes various methods for event extraction which are used in this research; section 4 explains experiments and their results and in the last section we conclude the paper and state future orientation.

2. Data and Material

We have used two textual datasets for training in this research. The first one is computer research center of islamic sciences, NOOR co. dataset which contains about 12000 die event paragraphs and its size is about 6.08 MB. This dataset contains texts from historical Arabic books such as Murujadh-dhahab, Al-Tabaghat Al-Kobra, Ya'qubi, Ansab Al-Ashraf and so forth. This dataset is formatted as XML file that provides related information about each paragraph such bellow:

```
<Title ID="1" Name="آدم (ع)" >
  <SubTitle Name="دفنهببيتالمقدس" >
     <ParagParag="854347" WordFrom="20"
      WordTo="24"
     Addr="تاريخ الطبرى ج١ ص:١٥٥-١٦١">
           حدثناابنسعد،قال: حدثناهشامقال: أخبرناابى، عنابىصالح، عنابنعباسقال:
                                          لماخرجنوحمنالسفينةدفنآدمعببيتالمقدس
     </Parag>
  </SubTitle>
           .
           .
           .
<Title>
```

It is worth to mention that we have only used the raw text for this research.

The second dataset is the text of 11 volumes of the great book of Al-Tabari, islamic history for training our model in other events (except death). The size of this dataset is 15.5 MB and has about 20000 sentences.

The test dataset for both above mentioned training sets is the text book of history of Ibn-Khaldoon vol.1. This dataset size is 1.85 MB and it has about 5000 sentences.

3. Proposed Approach

In this section, we describe required data preprocessing and our system architecture and their components.

3.1. Data Preprocessing

Data preprocessing is a necessary and time consuming step in the most IE systems. In this step data is converted to appropriate format required for IE process. The first step in our research is also raw text preprocessing.

Required raw text preprocessing in this research includes tokenizing, word stemming, Part Of Speech (POS) tagging, noun and verb group detecting (base phrase chunking) and name entity recognition. However, base phrase chunking and name entity recognition are only used in argument extraction and are not used in sentence classification as on-event or off-event.

We use AMIRAPOS Tagger [10] for tokenizing, POS tagging and base phrase chunking, Khoja [13] stemmer is also used to specify words stems. Also, some steps of the research require converting characteristic data to equivalent numeric values and we use Sally tool [19] for these. Noor ANER system implemented by Bidhendi *et al.* [8] is used for tagging name entities in the text with some settings for tagging places.

3.2. Event Extraction Methods

Event extraction includes two main stages: Event mention detection (sentence classification) and argument extraction. The following section describes the implementation of these stages.

3.2.1. Sentence Classification Stage

In this stage, sentences are classified as on-event or off-event. In this research, we use three methods including: Data oriented method, knowledge oriented method and semantic oriented method for sentence classification. Then, their performance and use are compared. In the following section we present more detailed descriptions of these methods.

3.2.1.1. Rule based Event Extraction

In this method, first a list of trigger keywords for each event was obtained by an expert team. The types of these keywords were noun, verb or other types of words which represent events. Then, based on trigger keywords and other linguistic conditions (such as POS tag and stem) various linguistic rules were applied to the text by using a hand coded program in order to extract event phrases and tag event mention. In this system, an event trigger may be presented by more than one word. In this case represented words may be separated by other words within the event mention.

3.2.1.2. Machine Learning based Event Extraction

We don't require expert linguistic knowledge and rule induction in this approach. At this step we implement a sentence classifier and multiple term classifiers using Support Vector Machine (SVM) classification method. SVM classifiers have proved to be robust in high volume text classification. A Set of used attributes for term classifiers includes:

- Words or Phrases of each Instance.
- Word Stem of each Word.
- POS Tag of each Word.

We run Library for SVM (LIB SVM) [16], which is a SVM classifier, in all possible combination of attributes. In other words the classifiers are taught using; all attributes, two attributes out of three and just one attribute which only contains the original words and later the results produced by the implementation of them on a set of tests are compared.

Then we use a SVM classifier for sentences classification as on-event or off-event sentences. SVM classifier trains on-event or off-event sentences using attribute values of train dataset. The Set of attributes in this case are as follows:

- Stem words in training dataset with a frequency greater than 10 are considered to be a set of features.
- Stem words with related POS tag as stem-POS in training dataset are other features, such as "موت_ VBD" which is a stem word of die event (موت) in the past time (VBD).
- "Status" Attribute which represents instance status as on-event or off-event.

Each sentence in train dataset is an instance that is configured as sequence of 0 and 1 value. Each word of an instance is compared with attributes set and if they are matched, value of 1 is assigned to the related column and otherwise 0 is assigned to it.

3.2.1.3. Semantic based Event Extraction

Due to positive impact of semantic methods in IE process, such methods have been applied. A lexical chain is a sequence of related words in writing, spanning short (adjacent words or sentences) or long distances (entire text) [15]. The use of lexical chain in event extraction process has been previously proposed by Sangeetha *et al.* [20] for English texts. Appling lexical chain to Arabic texts due to the lack of programming interfaces in order to communicate with Arabic word net and extract synonymous words is more difficult.

In this stage, event related lexical chains were automatically constructed by using Arabic word net [7] and training dataset sentences containing related events.

There are different algorithms to construct lexical chain; the method used in paper [20] with some changes was applied. In this method, we first preprocess sentences that contain desired event and applied POS tagger and remove stop words. Then, extract all synonyms, hypernyms and hyponyms of each word in sentences from word net. Synonyms are presented in the first level and hypernyms and hyponyms are presented in the second level. For each pair of candidate words, each sense of the word W_i is compared with all other senses of word W_j. If the match occurs, the words W_i, W_j and a list of matched words in the sense representation are included in the chain. Along with the matched words, path length with respect to the current level of matching and frequency of the word is also stored. Finally a lexical chain related to the desired event is selected. All of the above tasks were done automatically by a hand coded program.

To extract events by the lexical chain, each sentence in test dataset is searched for words in the lexical chain. Sentence score is calculated based on the number of words included in a lexical chain. Sentences with higher score are considered as more relevant to the desired event. In order to, determining threshold score value for finding relevant sentences, lexical chain was applied to train dataset which has only relevant sentences and then the lowest score of sentences was selected as the threshold score value.

One of the main advantages of using lexical chain in event extraction process is improving the automation of extraction so there is no need for an expert to determine keywords and rules (needs for knowledge based methods); also no need to create specified data format and determine and initialize key attributes (needs for machine learning methods). In this approach only by determining sentences containing desired event and applying lexical chain construction program, words relation has been formed and event related words have been placed in lexical chain. Then, we can apply this lexical chain to other texts and tag sentences containing event.

3.3. Argument Extraction Stage

Argument extraction is a complex and challenging task in an event extraction research. This stage requires further expert linguistic knowledge and complex rules. In this stage due to the high performance and existence of the knowledgeable team of experts, we use rule based sentence classification outputs for extracting arguments. We the nuse shallow parser modules for tagging grammatical relations in on-event sentences and finally extract each argument type using rule based methods. Our system architecture is shown in Figure 1.

- *Step* 1: Preprocessing.
- *Step* 2: Sentence Classification.
- *Step* 3: Subject-Verb-Object Detection.
- *Step* 4: Argument Extraction.

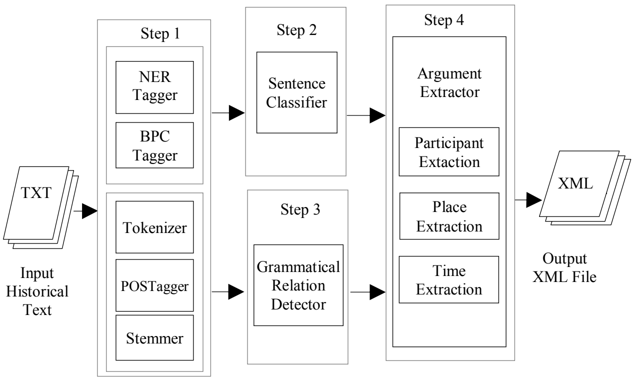

Figure 1. Event extraction system architecture.

3.3.1. Shallow Parsing

Shallow parsing module is required for extracting event argument using rule based methods. This module is a syntax analyzer and determines grammatical relations including subject-verb-object in on-event sentences. The inputs of our shallow parser module are text, noun, verb group tags and NER tags. This module uses a rule based method and maps grammatical rules like [11] to extract the triple subject, verb and object in each instance.

3.3.2. Rule-based Argument Extraction

Argument extraction stage uses a rule based method and prior step output to extract events argument according to ACE standard definition. In ACE standard some arguments are general event attributes which apply to the most events like TIME-ARG and PLACE-ARG which refer to the time and place of events respectively.

Some arguments are specific event attributes and apply to a specific event like VICTIM-ARG which is a specific argument for die event. In this system, each argument type performs the process of extraction independently and has specific rules, so lack of some information in common text sentences, do not have any effects on the following extraction process. The process of each argument type extraction is as follows:

- TIME-ARG Extraction: For extracting time role in an on-event instance, we first specify time phrases using time keywords and pattern matching technique. Then, we match the event time extraction rules to extract time role.
- PLACE-ARG Extraction: The Place of event is also a general argument like time which must be extracted in most event types. We use NER tags and event place extraction rules to specify event place. Also, as an additional way we use determinant particle like “فی” (“in” in English) which is followed by places (في “in”البصره+ ”Basra”) to specify places and increase extraction accuracy.
- Event Participants Extraction: We use syntactic labels created in the prior step and apply related rules to extract other event roles such as victim and agent. In case of the module failure due to prior step errors, we use NER and POS tags to specify sentence entities and map other rules to extract the related roles.

4. Experiments and Results

In this research we evaluate our experiment using performance measures metrics precision, recall and F-measure. We describe our experiment in the two main stages in the following.

4.1. Sentence Classification

As mentioned above we implement three methods for classifying sentences as on-event or off-event ones. In this section we introduce these systems implementation and their performance results.

To run the rule based event extraction system, a list of trigger keywords was prepared by the expert team. Using these keywords and determining related rules for desired event, phrases containing event was detected. This program has very good results. There are errors that have been often related to the prior steps in other words POS errors. Stemmer errors also cause some errors in event extraction stage. Table 2 shows the number of each event keywords.

Table 2. The number of trigger keywords set.

Event Type	Keyword Number
Die	30
Birth	7
Marriage	6
Divorce	6
Injury	3
Conflict	56

For instance, some sentences which contain event in training dataset in the Table 3 are presented. The underlined words are triggered words for die event in Arabic language, which our program applied related rules and recognized them and tagged these paragraph as an on-event paragraph.

Table 3. Examples of on-event sentences for die event in training dataset.

No	Example (Arabic)	Example (English)	Comment
1	حدثنا بكر عن ابن إسحاق قال: استشهد يوم اليرموك عمرو بن سعيد بن العاص و أبان بن سعيد بن العاص، و عكرمة بن أبي جهل، و عبد الله بن سفيان بن عبد الأسد، و سعيد بن الحارث بن قيس.	Bakr has told us of IbnIshaq that he said: Amribn Saeed ibn al-Aas and Abanibn Saeed ibn al-Aas, and IkrimahibnAbiJahl, and Abdullah bin SufyanibnAbd al-Assad, and Said ibn al-Harith bin Qais are martyred on Yarmouk day.	“استشهد” word means “martyrdom” which represents die event and our program tags this paragraph as on-event, because according to the applied rules, a verb with “شهد” stem in this linguistic format, represents a die event.
2	حدثني ابن كناسة عن الأشياخ قالوا: لما حضرت الأحنف الوفاة بالكوفة قال: لا تندبنّي نادبة و لا تبكيني باكية، و لا يعلمنّ بموتي أحد،	IbnKanasat has told me of elders that they said: when al-Ahnaf was dying in Kufa he said: do not wail and do not cry and do not tell anyone about my death.	“حضرت الوفاة” (although they are separated) represent die event; because when a verb with “حضر” stem is placed in sentences and after that with a limited distance words such as “موت” or “وفاة” are placed, die event is represented.
3	قال: و توفي محمد بن الفضيل بالكوفة سنة خمس و تسعين و مائة و شهد جنازته وكيع بن الجراح.	Said: Mohammad Ibn al-fazil died in Kufa in one hundred and ninety five year and saw his funeral Wakia bin al-jarah	There are two triggered words, although if one of them had occurred, it would have been enough for tagging on event paragraph.

As you have seen in the examples 2 and 3 in Table 3, this program does not rely on one word triggered or verb triggered. Phrases such as “حضرت الوفاة” (dying) are also considered as well as noun words such as “جنازة (funeral).

In the next step, we implemented SVM classifiers using the same dataset and different attribute types and numbers for term classification. Performance results for these classifiers presented in Table 4 show that the best result is for all feature based classifier. We used LIBSVM classifier which is a kind of SVM classifier. First we must have preprocessed dataset.

Table 4. Performance result of different applied systems for sentence classification for die event.

Algorithm	On-EventClass (%)			Off-EventClass (%)			Accuracy (%)
	Precision	Recall	F1	Precision	Recall	F1	
Keyword based Classification	96.15	84.75	90.09	98.15	99.58	98.86	97.96
SVM (Terms + Stem + POS)	96.67	71.19	82.35	96.58	99.79	98.16	96.85
SVM (Terms + POS)	95.45	71.19	81.55	96.57	99.54	98.03	96.66
SVM (Terms)	95.45	71.19	81.55	96.57	99.54	98.03	96.66
SVM (Sentences)	64.4	79.7	71.2	97.4	94.6	96	93.32
Lexical Chain based Classification	70.37	91.93	79.72	98.9	95.15	96.99	94.25

Each characteristic feature value must be converted into an equivalent numeric value in order to be processed by SVM. For this conversion sally tool was used and then by using a hand coded program each sample data together with its relevant attributes was converted to an appropriate LIBSVM classifier in order to build a proper model.

In generally LIBSVM input format is as follows:

*<Label><index*1*>:<value*1*><index*2*>:<value*2*>...*

Which *label* is the predicted value and other values are attribute values.

Following example is LIBSVM input format for word “توفى”:

1 1: *توفى* 2: *وفى* 3: *VBN_MS*3

Where *label* 1 is for the die event and the first value is the original word, the second value is the stem of the word and the last value is the POS of the word Having created the models by LIBSVM, we applied these models to test dataset which is Ibn-khaldoon history of Islam book vol.1 and tagged data as 1 (on-event) or 0 (off-event) which has above 80% F1-measure for die event see Table 4.

SVM Sentence classification is also, done with Sequential Minimal Optimization (SMO) algorithm which is provided by the Wekaframework [24].

We train model with 1562 features for die event which is train dataset words with frequency greater than 10 and their stem-POS features. Performances resulting from this case are shown in Table 4.

In the last stage for event sentence classification, we constructed lexical chain. For each event related lexical chain, we must use training documents related to the desired event.

We have firstly extracted lemma’s words using Buckwalter [9] software to search these words in the Arabic word net and extract synonyms, hyponym and hypernym of each word. Then, we applied our lexical chain construction program to create documents lexical chains. Then, among these lexical chains, we selected that lexical chain which is related to the desired event. This lexical chain after removing low frequency and high path length items applied to test dataset is tagged on event sentences.

We have constructed lexical chain using both original words and stemmed words and compared their performance. Using stemming mode we can reach more efficient performance. In this mode, the words of training documents have been stemmed and later we applied lexical chain construction program.

Also, in testing process, firstly all of the words have been stemmed and then have been compared with event related lexical chain words. Comparing lexical chain words and words of test sentences, each sentence gives a score according to the shared words. More shared words are equal to more probability for describing the event in the sentence. In this research, for determining threshold score value to find relevant sentences; we apply lexical chain to train dataset which has only relevant sentences and lowest score of sentences selected as threshold score value. Performance result has been shown in Table 4.

As the results show, the best result of sentence classification based on F-measure was for Keywords based classification. SVM with all features (terms+stem+POS) has also high results, that if there is required domain knowledge for knowledge oriented approaches or large amount of data and required preprocessing for data oriented approach, these methods are the best choices. But, lexical chain based method has an acceptable precision about 70% and high recall about 92% for detecting on-event sentences; Advantages of using lexical chain is that in this case there is no need to high domain knowledge or large amount of data as well as preprocessing phase and formed attribute values. This method is the best choice in many situations.

4.2. Argument Extraction

For extracting argument we use Computer Research Center of Islamic Sciences, NOOR co. dataset which is a large volume dataset containing about 12000 die event paragraphs. Several events may occur in one paragraph. Performance results for argument extraction module in 420 paragraphs are shown in Tables 5, 6 and Figure 2. This module extracts argument with overall precision 80.51 and recall 71.29. However, a large number of errors in this stage belong to the prior step errors like POS tagger, stemmer and NER tagger that have affected our results.

Table 5. Performance results in argument extraction step for die event.

Argument	Precision (%)	Recall (%)	F-measure (%)
Agent	88.64	70.27	78.89
Victim	81.57	64.81	72.23
Place	57.3	42.5	48.8
Time	81.57	84.12	82.82

Table 6. Average and overall performance results in argument extraction step.

	Precision (%)	Recall (%)	F-measure (%)
Average Results	77.27	65.42	70.85
Overall Results	80.51	71.29	75.62

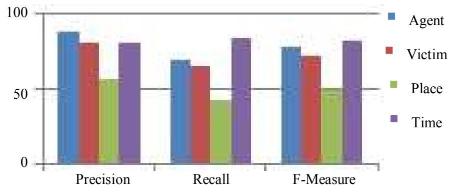

Figure 2. Comparing performance results in argument extraction process.

System implementation and rule mapping was done using C# programming. Some output instances of argument extraction module are shown in Table 7. Underlined words are extracted information. As it is shown in example 4 in Table 7, one argument such as victim in the above example can consist of several instances. Also in this example, time argument consists of two parts that one part is at the beginning and another one is at the end of the sentence.

Table 7. Examples of argument extraction outputs for die event.

No	Example (Arabic)	Example (English)	Trigger	Victim	Agent	Place	Time
1	و في السنة السادسة أخرجته أمه إلى أخواله زائرة فتوفيت بين مكة و المدينة،	In the sixth year his mother took him to his maternal uncles while going on a pilgrimage. Then died Between Mecca and Medina	توفيت	أمه	-	بين مكة و المدينة	السنة السادسة
2	الربعي. توفي بالكوفةفي خلافة أبي جعفر. و عيسى بن موسى وال على الكوفة.	Al-Reb`i. died in Kufa in the succession of AbiJaafar. And Isa ibn Musa was the governor of Kufa.	توفي	الربعي	-	بالكوفة	في خلافة أبي جعفر
3	قال أبو الحسن: أبان بن سعيدقتليوم أجنادين، و يقال: يوم مرج الصفر.	Abu Hassan said: Aban bin Saeed was killed in Ajnadeen day that is said: Marj al-Safar day	قتل	أبانبن سعيد	-	-	يوم أجنادين و يقال : يوم مرج الصفر
4	و في خلافة المتوكلماتهدبة بن خالد، و شيبان بن فروخ الأبلي، و إبراهيم بن محمد الشافعي، و ذلك في سنة ست و ثلاثين و مائتين	In succession MutawakilHadba bin Khalid, and Shiban bin FarroukhAlabla, and Ibrahim bin Mohammed Al-Shafi'i died in two hundred and thirty six years	مات	هدبة بن خالد ، شيبان بن فروخ الأبلي، إبراهيم بن محمد الشافعي			سنة ست و ثلاثين و مائتي في خلافة المتوكل
5	فيها ماتحرمي بن حفص، و محمد بن كثير، و موسى بن إسماعيل و أبو عبد الله الخزاعي، و أبو أمية، و أبو صالح كاتب الليث.	where, Harmi bin Hafs, and Mohammad bin Kathir, and Musa bin Ismail and Abu Abdullah al-Khuzai, and Abu Omaya, and Abu Saleh Laith writer died	مات	حرمي بن حفص، محمد بن كثير، موسيين إسماعيل، أبو عبد الله الخزاعي و أبو أمية أبو صالح			
6	و فيها ماتابو عمر الزاهد، غلام ثعلب، و جوز العالم جنازته فيالكرخ، فوقعت الفتنة لأجلها	And where Abu Omar al-ZahedThalab`s Servant died and his corpse Fell into the Karkh. Then Fell sedition for this.		مات جنازته	Connected pronoun reference of ”جنازته“ (ابوعمرالزاهد)		
7	حدثنا علي بن عاصم عن حصين عن عمرو بن جاوان عن الأحنف قال: لما انحاز الزبيرقتلهعمرو بن جرموز بوادي السباع.	Ali ibnAsim told us of Hasin of Amr bin Jawan of Ahnaf that he said: When Zubair was defeated, AmribnJermoz killed him in a wide animal place	قتله	مرجع ضمير متصل به قتله(الزبير)	عمرو بن جرموز		

Also there is only a victim role and other roles don’t appear in the sentence in example 5.

Our system is able to detect pronoun reference such as victim in the example 6. Also, a sentence may include several event instances which all of them are extracted with corresponding argument.

In example 7 place argument is not specified. This error is caused by NER error in tagging this place as place entity and to kenizer error in separating some prepositions like “ب” which are located before “ وادي السباع” in the above example.

5. Conclusions and Future Work

Due to the wide use of event extraction in many natural languages processing applications, there are high research needs in this area.

In this research we implemented an event detection and extraction system for the classical Arabic historical texts.

The linguistic complexity of the Arabic language and different structures of historical texts are the research challenges.

We could not compare this research results with other studies, because it is the first one in historical Arabic texts and duo to different linguistic characteristics of modern Arabic texts and historical ones comparing these is not true. But we have compared different methods results which used in this research.

In this paper, we used different event extraction approaches including knowledge oriented, data oriented and semantic oriented approaches to sentence classification as on-event or off-event in Arabic documents. Based on prerequisites and required conditions of each approach implementation and execution, expected accuracy and precision, we can select the best one for the event extraction.

Knowledge oriented method requires linguistic rules and trigger keywords and leads to high precision and recall values.

Data oriented method, which is a SVM classifier, requires a large amount of data and many data preprocessing to create models and can produce high efficient results.

Semantic oriented method, which includes lexical chain construction and applying it to documents, does not need high domain knowledge, large amount of data and many data preprocessing and can reach an acceptable precision and high recall for the event extraction. Figures 3 and 4 show performance results for different event types.

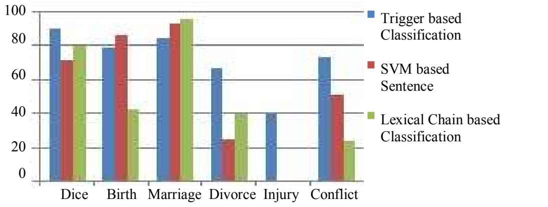

Figure 3. Comparing all methods f1 (%) of on-event class for six event type.

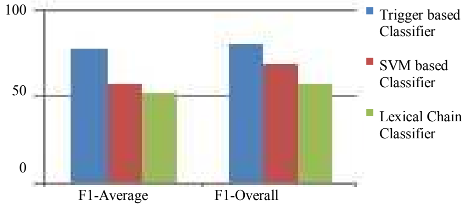

Figure 4. Average and overall f1 of on-event class for six event type.

In the next stage, our system uses rule based approach and extracts information based on lexicosyntactic rules. System architecture includes a modular structure. Event information is extracted independently, so extraction failure in one element does not affect another one.

As future work, we can increase attribute number in SVM model and evaluate performance of SVM based classifier.

To improve lexical chain method, we can involve other factors, (in addition to the shared word number between the lexical chain and the processing sentence) to tag event sentences, like the time of verb words and POS of words in training document words and to apply them as impacted factor in the testing process.

For argument extraction, we can create rules automatically using rule induction methods and few primary rules. In such systems, requirements to the expert linguistic knowledge will be less.

Acknowledgement

We are thankful to Noor Text Mining Research Center which is affiliated to Noor Computer Research Center of Islamic Sciences located in Qom. They gave their historical corpora to us for this research. We extend our grateful thanks to Mr. HoseinJouzi and Mr. HabibzadehBijani who helped us in providing historical texts as well as evaluating the enrichment phase of the historical Islamic texts specified for this research.

References

[1] Abdul Halin A., Rajeswari M., and Abbasnejad M., “Soccer Event Detection via Collaborative Multimodal Feature Analysis and Candidate Ranking,” *the International Arab Journal of Information Technology*, vol. 10, no. 5, pp. 493-502, 2013.

[2] Abuleil S., “Using NLP Techniques for Tagging Events in Arabic Text,” *in Proceedings of the 19th International Conference on Tools with Artificial Intelligence*, Patras, Greek, pp. 440-443, 2007.

[3] ACE Overview., available at: http://projects.ldc.upenn.edu/ace/intro.html, last visited 2012.

[4] Ahn D., “Stage of Event Extraction,” *in Proceedings of Workshop on Annotating and Reasoning about Time and Events*, Sydney, Australia, pp. 1-8, 2006.

[5] Ananiadou S., Pyysalo S., Tsujii J., and Kell D., “Event Extraction for Systems Biology by Text Mining the Literature,” *Trends in Biotechnology*, vol. 28, no. 7, pp. 381-390, 2010.

[6] Aone C. and Ramos-Santacruz M., “REES: A Large-Scale Relation and Event Extraction System,” *in Proceedings of the 6th Conference on Applied Natural Language Processing*, Washington, USA, pp. 76-83, 2000.

[7] ArabicWordNet., available at: http://www.globalwordnet.org/AWN/, last visited 2012.

[8] Bidhendi M., Minaei-Bidgoli B., and Jouzi H., “Extracting Person Names from Ancient Islamic Arabic Texts,” *in Proceedings of Language Resources and Evaluation for Religious Texts Workshop*, Istanbul, Turkey, pp. 1-6, 2012.

[9] Buckwalter., available at: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49, last visited 2012.

[10] Diab M., “Second Generation AMIRA Tools for Arabic Processing: Fast and Robust, Tokenization, POS tagging and Base Phrase Chunking,” *in Proceedings of the 2nd International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, pp. 285-288, 2009.

[11] Hammadi O. and Ab Aziz M., “Grammatical Relation Extraction in Arabic Language,” *the Journal of Computer Science*, vol. 8, no. 6, pp. 891-898, 2012.

[12] Hogenboom F., Frasincar F., Kaymak U., De Jong F., “An Overview of Event Extraction from Text,” *in Proceedings of Detection, Representation and Exploitation of Events in the Semantic Web*, Bonn, Germany, pp. 48-57, 2011.

[13] Khoja S., available at: http://zeus.cs.pacificu.edu/shereen/research.htm, last visited 2012.

[14] Lei B. and Sheng B., “Methods of Customer Requirements Feature Extraction on Product Reviews,” *the Journal of Information and Computational Science*, vol. 9, pp. 2429-2439, 2012.

[15] Lexical Chain., available at: http://en.wikipedia.org/wiki/Lexicalchain, last visited 2012.

[16] LibSVM., available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/, last visited 2012.

[17] Naughton M., Stokes N., and Carthy J., “Investigating Techniques for Sentence-Level Event Classification,” *in Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, UK, pp. 617-624, 2008.

[18] Piskorski J., Tanev H., and Wennerberg P., “Extracting Violent Events from on Line News for Ontology Population,” *in Proceedings of the 10th International Conference on Business Information System*, Poznan, Poland, pp. 287-300, 2007.

[19] Sally, available at: http://mloss.org/revision/view/960/, last visited 2012.

[20] Sangeetha S., Takur R., and Arock M., “Event Detection using Lexical Chain,” *in Proceedings of the 7th International Conference on Natural Language Processing*, Reykjavik, Iceland, pp. 314-316, 2010.

[21] Sangeetha S., Takur R., and Arock M., “Domain Independent Event Extraction System using Text Meaning Representation Adopted for Semantic

Web," *the International Journal of Computer Information Systems and Industrial Management Applications*, vol. 2, pp. 252-261, 2010.

[22] Vargas-Vera M. and Celjuska D., "Event Recognition on News Stories and Semi-Automatic Population of an Ontology," *in Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence*, California, USA, pp. 615-618, 2004.

[23] Wayne C., "Topic Detection Tracking (TDT)," *in Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, Maryland, USA, pp. 1-3, 1998.

[24] Witten L. and Frank E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 2000.

[25] Xu F., Uszkoreit H., and Li H., "Automatic Event and Relation Detection with Seeds of Varying Complexity," *in Proceedings of the AAAI Workshop Event Extraction and Synthesis*, Massachusetts, USA, pp. 12-17, 2006.

Razieh Baradaran received her BSc and a MSc degrees in the Department of Information Technology at University of Qom, Iran in 2010 and 2013 respectively. Her research interests include: Information extraction, text mining and intrusion detection systems.

Behrouz Minaei-Bidgoli obtained his PhD degree from Michigan State University, East Lansing, Michigan, USA, in the field of data mining and web-based educational systems in computer science and engineering department. He is working as an assistant professor in Computer Engineering Department ofIran University of Science and Technology, Tehran, Iran. He is also leading at a Data and Text Mining research group in Computer Research Center of Islamic Sciences, NOOR co. Qom, Iran, developing large scale NLP and Text mining projects for farsi and arabic languages.