Current Trends in Technology and Science
ISSN : 2279-0535. Volume : III, Issue : I

# An NLP Based Approach for Extracting Intelligence from event documents

**Name : Prashant G. Desai**
Designation : Lecturer in Computer Science Department, Organization :N.R.A.M.P.,Nitte-574110, Karnataka, India, Email ID : prashanth_desai@yahoo.com

**Name : Sarojadevi H.**
Designation : Professor in Computer Science Department, Organization :N.M.Institute of Technology, Bengaluru, Karnataka, India, Email ID : hsarojadevi@gmail.com

**Name : Niranjan N. Chiplunkar**
Designation : Principal, Organization :N.M.A.M. Institute Of Technology, Nitte-574110, Karnataka, India, Email ID : nirnjanchiplunkar@rediffmail.com

***Abstract*** **- A vast amount of electronic information is available in the form of documents such as papers, emails, reports, html pages etc. Sifting through such documents can result in very essential information. An automated tool would be of great use, for identifying and extracting this kind of information. This paper presents an automated approach for identifying a set of event patterns called intelligent information from natural language text.**

***Keyword*** **- Automation, data mining, events, information extraction, natural language.**

## 1. INTRODUCTION

Information is at the core of everything around us. Our entire existence is a process of gathering, analyzing, understanding, and acting on the information. To empower people who are in need of information, technologies like data warehousing, metadata repositories, online analytical processing (OLAP) and data mining have emerged [1]. Of these technologies data mining is the process of extracting useful information. Information extraction is a method of finding interesting and very essential patterns of text. One common variation of information extraction is event extraction which encompasses deducing specific intelligence referred to in text [2].

## 2. EVENT EXTRACTION

Frederic et. Al. [2] distinguishes between three main approaches to event extraction.

### 2.1. Data driven event extraction

Data-driven text mining approaches use basic statistical reasoning based on probability theory, all quantitative approaches to automated language processing, such as probabilistic modeling, information theory, and linear algebra. Despite their differences, all approaches focus on discovering statistical relations, i.e., facts that are supported by statistical evidence. Examples of discovered facts are words or concepts that are (statistically) associated with one another. However, statistical relations do not necessarily imply semantically valid relations, or relations that have proper semantic meaning.

### 2.2. Knowledge driven data extraction

In contrast to data-driven methods, knowledge-driven text mining is often based on patterns that express rules representing expert knowledge. It is inherently based on linguistic and lexicographic knowledge, as well as existing human knowledge regarding the contents of the text that is to be processed. Patterns are useful when one needs to extract very specific information. However, in order to be able to define patterns that retrieve the correct, desired information, lexical knowledge and possibly also prior domain knowledge is required.

### 2.3. Hybrid event extraction

As both approaches have their disadvantages, in practice combining two methods could yield the best results. In general, there is an increasing number of researchers that equally combine both approaches, and thus employ hybrid approaches.

## 3. RELATED WORK

Authors of [3] propose a cross-document event extraction system. They first apply ACE single document IE system which can extract events from individual documents. This IE system includes entity extraction, time expression extraction and normalization, relation extraction and event extraction. Those person entities involved frequently in events with high confidence are labeled as centroid entities. Authors first construct the candidates through a simple form of cross-document co reference and then rank these candidates. Authors exploit knowledge derived from the background data (related documents and Wikipedia) to improve performance.

[4] provides an update on the “Artequakt” system which uses natural language tools to automatically extract knowledge about artists from multiple documents based on a predefined ontology. The Artequakt project has

implemented 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. The aim of the knowledge extraction tool of Artequakt is to identify and extract knowledge triplets (concept – relation – concept) from text documents and to provide it as XML files for entry into the KB.The extraction process is launched when the user requests a biography for a specific artist that is not in the KB. A script was developed to query the artist's name in general-purpose search engines, such as Google and Yahoo. Documents returned by the search engines need to be filtered to remove irrelevant ones. GATE is used for entity recognition. Word Net is used to support relation extraction. Each selected document is divided into paragraphs and sentences. Each sentence is analyzed syntactically and semantically to identify and extract relevant knowledge triples. [5] gives an overview of the Real-time News Event Extraction Framework developed for Frontex (the European Agency for the Management of Operational Cooperation at the External Borders of the Member States of the European Union), to facilitate the process of extracting structured information on border security related events from on-line media in order to support situation monitoring and intelligence gathering, with a particular focus on incidents related to illegal migration (e.g., illegal entry attempts), cross-border crime (e.g., smuggling), and crisis situations (e.g., violent events, natural disasters, biohazards) at and beyond the EU external borders. First, news articles are gathered by a large scale news aggregation engine, the Europe Media Monitor (EMM). EMM retrieves more than 100,000 news articles per day from more than 2500 news feeds in 42 languages. These news articles are geo-located, tagged with meta-data and further filtered using standard keyword-based techniques in order to select those articles, which potentially refer to security-related incidents and events. In addition, the news articles harvested within a 4-hour window are grouped into clusters in every language individually according to content similarity (using hierarchical agglomerative clustering). The filtering and clustering process is performed every 10 minutes. Next, the stream of filtered news articles and clusters is passed every 10 minutes to the event extraction engine, which consists of two core event extraction systems, namely, NEXUS and PULS. NEXUS follows a shallow cluster-centric approach. Each cluster of topically related articles undergoes a shallow linguistic analysis, i.e., fine-grained tokenization, morphological analysis, gazetteer look-up, sentence boundary detection, etc., and a cascade of simple finite-state extraction grammars is applied to each article in the cluster. While the lower-level grammars are used to extract person names (e.g., George Bush), person groups (e.g., Algerian immigrants), numerical expressions (e.g., two hundred), quantifiers (e.g., More than), and other small-scale structures.

## 4. PROPOSED APPROACH

An input document will be divided into paragraphs and sentences. Each sentence will be analyzed syntactically and semantically to identify and extract important events. Figure 1 presents the overall procedure of the event extraction process.

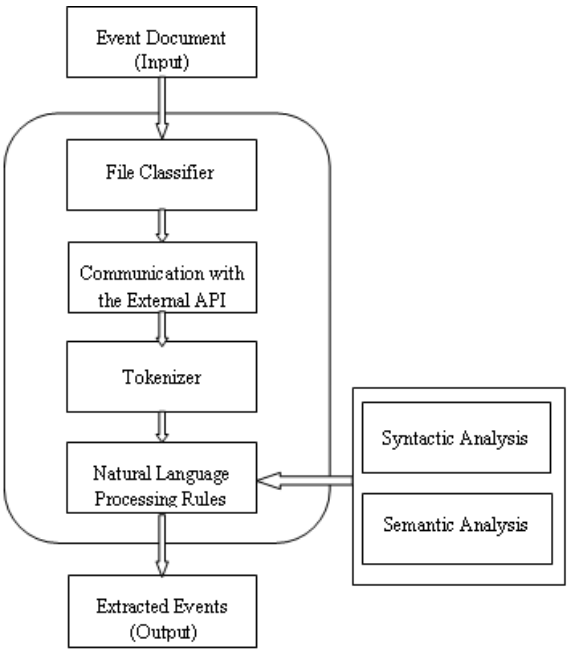


**Figure 1 System architecture**

### 4.1 File classifier

The proposed work is implemented in such a way that user can provide Ms-Word or pdf format of input document. This module identifies whether it is a MS-Word or pdf document.

### 4.2 Communication with the external system

This module is responsible for communicating with the external tools used while implementing our work. We use mainly two tools PDFOne [6] and Apache POI [7].

#### 4.2.1 PDFOne

Gnostice PDFOne is a versatile PDF SDK for implementing PDF-related features in Java applications. PDFOne can create, edit, view, print, encrypt, decrypt, merge, split, reorganize, bookmark, annotate, watermark, and stamp PDF documents. The API hides the complexity of the PDF format and enables to quickly implement sophisticated PDF features. PDFOne is entirely written in Java code.

#### 4.2.2 Apache Poor Obfuscation Implementation (POI)

This is a Java API to Handle Microsoft Word Files. HWPF is the port name of Microsoft Word 97(-2007) file format to Java. Word document can be considered as

very long single text buffer. HWPF API provides "pointers" to document parts, like sections, paragraphs and character runs. A Word file is made up of the document text and data structures containing formatting information about the text. The entry point for HWPF's reading of a Word file is the File Information Block (FIB). This structure is the entry point for the locations and size of a document's text and data structures. The FIB is located at the beginning of the main stream.

### 4.3 Tokenizer

Tokenization is a process of breaking input stream of characters into an ordered sequence of words like units, usually called tokens that can be used for further processing [8]. These tokens may correspond to words, numbers, and punctuation marks. Tokenization is a prerequisite in order to perform any Information extraction task.

### 4.4 NLP rules

These rules are used to syntactically and semantically analyze the input document.

#### 4.4.1 Syntactic analysis

It is a process of grouping of words without referring to the context of their appearance in the text.

#### 4.4.2 Semantic Analysis

First this module identifies the boundary of sentence and then breaks the text of whole document into sentences. Then each sentence is broken down into words. Next step is to identify events. Identification of events is done by processing every word in each sentence. Presently, our work is focused on extracting important dates from the event documents such date for registration, paper submission date, date of acceptance etc.

## 5. Evaluation Metrics

Josef Steinberger et. al. [9] have proposed few parameters for measuring text quality which are often assessed by human annotators. Few aspects of text quality are listed below.

**Grammaticality** – the text should not contain non-textual items (i.e., markers) or punctuation errors or incorrect words.

**Non-redundancy** – the text should not contain redundant information reference.

**Clarity** – the nouns and pronouns should be clearly referred to in the summary. For example, the pronoun he has to mean somebody in the context of the summary.

## 6. Results and Analysis

This section describes performance and results obtained out of the system implemented. The system described is tested on ten event documents having different words count to extract intelligence from the text.

**Coherence and structure** – the summary should have good structure and the sentences should be consistent.

Many human language technology (HLT) tasks such as Information extraction are traditionally evaluated using Precision, Recall and F-measure [10]. We have evaluated results obtained based on these three metrics.

### 5.1. Precision

Precision measures the number of correctly identified items as a percentage of the number of items identified. In other words, it measures how many of the items that the system identified were actually correct, regardless of whether it also failed to retrieve correct items. The higher the Precision, the better the system is at ensuring that what has been identified is correct. It is formally defined as

$$\text{Precision (P)} = \frac{\left(\text{Correct} + \frac{1}{2}\text{Partial}\right)}{\text{Correct} + \text{Spurious} + \text{Partial}}$$

### 5.2. Recall

Recall measures the number of correctly identified items as a percentage of the total number of correct items. In other words, it measures how many of the items that should have been identified actually were identified, regardless of how much spurious identification was made. The higher the Recall rate, the better the system is at not missing correct items. Recall is formally defined as

$$\text{Recall (R)} = \frac{\left(\text{Correct} + \frac{1}{2}\text{Partial}\right)}{\text{Correct} + \text{Missing} + \text{Partial}}$$

**5.3. F-measure** The F-measure is often used in conjunction with Precision and Recall, as a weighted average of the two. If the weight is set to 0.5 (which is usually the case), Precision and Recall are deemed equally important. F-measure is formally defined as

$$\text{F} - \text{measure} = \frac{\beta^2 + 1 \quad \text{P} * \text{R}}{\beta^2 \text{R} + \text{P}}$$

Where β reflects the weighting of P vs. R. If P and R are to be given equal weights, then we can use the equation

$$\text{F1} = \frac{P * R}{0.5 * \text{P} + \text{R}}$$

### 6.1. Experimental setup

Initially a file is input to the system. Based on the file extension the file classifier identifies whether the input file is a MS-Word document or “pdf” document. If it is MS-Word document then the file classifier feeds file into Poor Obfuscation Implementation File System (POIFS)

subcomponent of Apache tool otherwise file is transferred to PDFOne component for reading "pdf" document.

**6.2 Processing a Word File**

1. An instance of POIFSFileSystem is created using an input stream from which data is to be read.
2. The input stream is read till EOF.
3. The constructor HWPFDocument which uses POIFSFileSystem containing the Word document as parameter is used for loading the Word document.
4. Text from the loaded word document in step-3 is extracted as paragraphs using constructor WordExtractor for further processing.

**6.3 Processing a pdf file**

1. An instance of PDFOne class is used to create a new pdf document from scratch and load an existing document.
2. The lines of text from the loaded document are then extracted for further processing.

**6.4 Identification of dates**

1. Break text from document into lines.
2.For each line of text in the document
    If dates are present then
        Output current line of text
    End if
End for

**6.5. Tabulation of evaluation metric values**

We evaluated performance of the system and resulting values for precision, recall and F-measure as well as average time taken for executing each document are shown in Table 1.

**6.6. Results obtained**

Figure 2 illustrates performance results. It can be seen that performance remains unaffected even if size (number of words) of the document increases. This same observation forecasted using a trend line in figure 2. However, variation in the graph is due to structure of the document like document containing tables. Figure 3 and figure 4 present snap shots of our system.

| Sl. No. | Length Of File (Number Of Words) | Correct | Partial | Spurious | Missing | Precision (P) | Recall ( R ) | F-Measure (F1) | Average Execution Time (Sec) |
|---|---|---|---|---|---|---|---|---|---|
| 1. | 139 | 2.00 | 1.00 | 0.00 | 1.00 | 0.83 | 0.63 | 0.71 | 18.33 |
| 2. | 149 | 4.00 | 1.00 | 0.00 | 0.00 | 0.90 | 0.90 | 0.90 | 20.53 |
| 3. | 167 | 2.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 17.94 |
| 4. | 168 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 17.44 |
| 5. | 172 | 3.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 19.46 |
| 6. | 184 | 2.00 | 1.00 | 0.00 | 0.00 | 0.83 | 0.83 | 0.83 | 14.37 |
| 7. | 187 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.75 | 0.75 | 11.35 |
| 8. | 212 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 17.33 |
| 9. | 348 | 4.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 27.76 |
| 10. | 367 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 25.25 |

**Table 1 Statistics of the evaluation results**

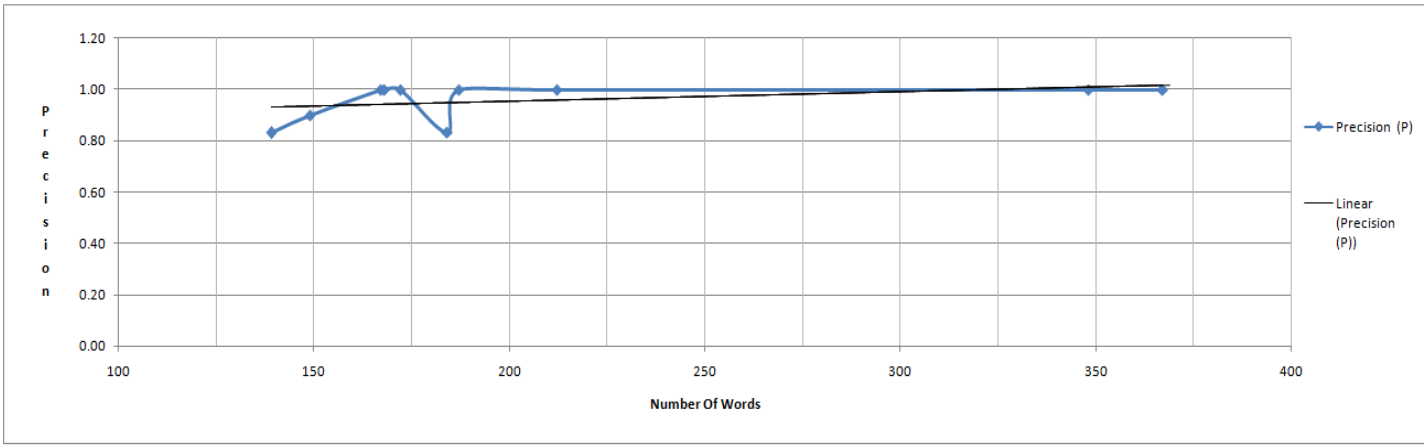


**Figure 2 Performance dependency on length of document**

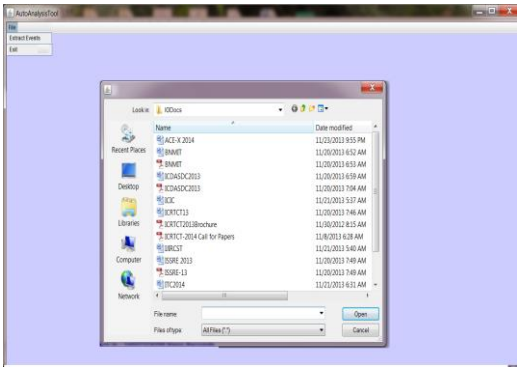

**Figure 3 Input file selection**

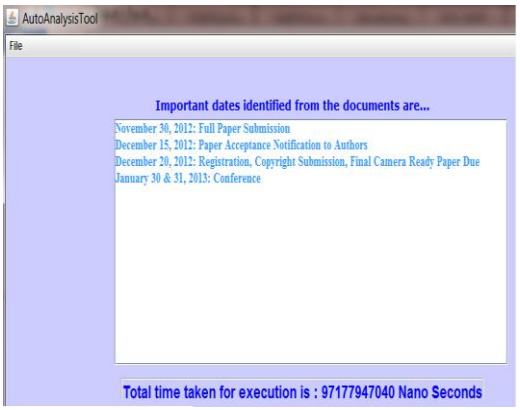


**Figure 4 Output**

## 7. Conclusion

This paper has presented a work which can be used for extracting intelligence from the event documents. Presently, we have used conference brochures as event documents. Conference documents contain several important dates pertaining to various events. The system is able to identify the important dates successfully. The experimental results demonstrate the validity of the approach followed by us while extracting the information.

## 8. References

[1] Alex Berson, "Data ware housing, data mining and OLAP", Tata McGraw Hill, 2011

[2] Frederick hogenboom, Flavius Frasincar, Uzay Kaymak, Franciska de Jong, "An overview of event extraction from text", Proceedings of 1st workshop on detection, representation & exploration of events, ISWC, 2011

[3] Heng Ji, Ralph Grishman Zheng Chen, Prashant Gupta, "Cross document event extraction and Tracking: task evaluation techniques and challenges", Proceedings of ICRANLP, pp. 166–172, 2009

[4] Harith Alani, Sanghee Kim, David E Millard, Mark J Weal, Paul H Lewis, "Automatic extraction of Knowledge from web documents", Proceedings of IEEE Intelligent Systems, 2006

[5] Martin Atkinson, Jakub Piskorski, "Frontex real-time news event extraction framework", Proceedings of 17th ICKDDM, ACM, pp. 749– 752,2011

[6] "http://www.gnostice.com/PDFOne_Java.asp"

[7] "http://poi.apache.org/hwpf/index.html"

[8] Jakub Piskorski, "Core Linguistic entity online extraction", JRC Scientific and Technical Reports, 2008

[9] Josef Steinberger, Karel Jezek, "Evaluation measures for text summarization", Proceedings of computing and informatics, Vol. 28, 2009

[10] Dayana Maynard, Wim Peters, Yaoyong Li, "Metrics for evaluation of ontology based information extraction", Proceedings of WWW workshop, 2006