*Information Technologies, Management and Society, 2014, Volume 7, No. 1, 21 - 28*

# INFORMATION EXTRACTION FROM THE THEORETICAL PERSPECTIVE

**M.T. IPALAKOVA**

*International University of Information Technologies*
*Department of Information Technologies*
*Manas Street 34A,050040, Almaty, Kazakhstan,*
*e-mail: m.ipalakova@iitu.kz*

**ABSTRACT**

In this article the information extraction field of study is discussed. Such aspects like its place in the text mining pipeline, the definition, the history of establishing the evaluation process are considered in detail. The comparisons of two extraction approaches and architectures for the information extraction systems design are presented.

**Key words:** information extraction, information extraction systems, entities, events, relations, knowledge engineering and machine learning approaches, precision, recall, evaluation conferences.

## 1. INTRODUCTION

Nobody will contradict the statement that we live in the Information Age. Information *per se*, information technologies in general, the Web and the Internet in particular have totally changed the way we work, study and communicate. There is an enormous amount of information on the Web which is available now for almost anyone. According to Moens [13] there have been several attempts to estimate how much information the Web contains. Even though it is obvious that such kinds of measurements are very rough and approximate, they allow us to gain general understanding of the volume of available data and predict that if the trend remains the same we will have to estimate the information in yottabytes (1 yottabyte is equal to $2^{80}$ bytes) in the near future.

However, the amount of accessible information would not be of much use if there were no suitable techniques to process it and extract knowledge from it. Thus, text mining is one of the technologies which are employed for those purposes. It can be described as a process of identifying the unknown information from a variety of unstructured data sources with a goal of further analysis of the derived facts.

In the context of text mining technology information extraction can be classified as one of the pre-processing tasks which are used in order to make data ready for applying major text mining techniques. These pre-processing operations involve processing the input, unstructured information in the form of documents, and presenting it in a more structured way to make further post-processing analysis possible.

## 2. DEFINING INFORMATION EXTRACTION

Despite the fact that information extraction is generally considered as a link in the chain of text mining techniques, it is a powerful and self-dependent technology.

There are a lot of situations when information must be analysed somehow but it is available primarily only in the form of natural text, such as technical reports, scientific articles, log records, news,

etc. For instance, a hospital wants to produce its own statistics about the most commonly encountered diseases within the age and gender groups of patients. But the data they need is mostly stored in medical records in textual form. Another example can be provided from a business area. A particular company or business agency wishes to know the tendency of enterprises' bankruptcies by industries. That kind of information can be taken only from news reports. In both cases the information extraction is able to help and accomplish those kinds of tasks avoiding people to process large amounts of text documents by hand. It reduces the amount of information to be analysed by extracting useful facts and ignoring irrelevant ones. Derived data is presented then in a more structured database way when it is easily accessible for applying different analysing techniques [9, 10].

In order to explain the term information extraction, definitions from the different authors, namely Moens [13], Cowie and Lehnert [2], Grishman [9], Turmo *et al*. [14], have been examined. As soon as there is no classical definition for information extraction every author defines it in the way which he or she believes explains information extraction in the better way. Here is the definition of information extraction we have come up with taking into account the considerations of the authors mentioned above. It is defined in a more simplified way but without losing its core idea and aims. Information extraction is the identification and selection of the named entities relevant to the specific task, of the relationships between them and events in which they participate in the natural language text in order to make them more accessible for further manipulations.

## 3. THE OVERALL PROCESS OF INFORMATION EXTRACTION

Analysing the process of information extraction it is become obvious that different authors divide it in different steps of different granularity, combining them into bigger stages and assigning the components of the information extraction systems to accomplish the tasks involved (1, 2, 6, 9, 11, 14]. However, analysing those different approaches the general pipeline of the information extraction process can be summarised and six main stages can be determined as following:

1. **Initial processing** which includes splitting a text into the fragments which are defined like zones, sentences, segments or tokens. This procedure can be performed by the components named as tokenisers, text zoners, segmenters or splitters. As Appelt and Israel [1] stated, tokenisation is a quite straightforward task for the texts in any European language, where the blank space between characters and punctuation indicate the boundaries of a word and a sentence respectively. But, for example, for Chinese or Japanese texts, where the boundaries are not so obvious this operation is not the simple one and requires much more effort to fulfill it. The next task within this stage is usually the morphological analysis which includes part-of-speech tagging and phrasal units (noun or verb phrases) identification. Part-of-speech tagging might be helpful to the next step which is the lexical analysis. It handles unknown words and resolves ambiguities. In addition, the lexical analysis involves working with the specialised dictionaries and gazetteers, which are composed of different types of names: titles, countries, cities, companies and their suffixes, positions in a company, etc. If a word in a document is found in a gazetteer it is tagged with the semantic class the word belongs to. For example, a word "Mr" will be tagged with the semantic class "Titles" [11, 14].

2. **Proper names identification** (names of people or organisations, dates, currency amounts, locations, addresses, etc.) is one of the most important operations in the chain of information extraction. Proper names can be encountered in almost all types of texts and usually they constitute the part of the extraction scenario. These names are recognised using a number of patterns which are called regular expressions [6].

3. **Parsing**. During this stage the syntactic analysis of the sentences in the documents is performed. After the previous step, where the basic entities were recognised the sentences are parsed to identify the noun group around some of those entities and verb groups. This parsing stage must be done in order to prepare the ground for the next stage of extraction relations between those entities and events in which they participate. The noun and verb groups are used as sections to begin to work on at the pattern

matching stage. The identification of those groups is realised by applying a set of specially constructed regular expressions [6, 9].

4. **Extraction of events and relations**. Everything which is done previously is basically the preparation for the major stage of extraction of events and relations, which are particularly related to the initial extraction specifications given by a client. This process is realised by creating and applying extraction rules which specify different patterns. The text is matched against those patterns and if a match is found the element of the text is labelled and later extracted. The formalism of writing those extraction rules differs from one information extraction system to another [1, 6, 9].

5. **Anaphora resolution**. Any given entity in a text can be referred to several times and every time it might be referred differently. In order to identify all the ways used to name that entity throughout the document coreference resolution is performed. Coreference or anaphora resolution is the stage when for noun phrases it is determined if they refer to the same entity or not. The most common types are pronominal and proper names coreference, when a noun is replaced by a pronoun in the first case and by another noun or a noun phrase in the second one [1, 6].

6. **Output results generation**. This stage involves transforming the structures which were extracted during the previous operations into the output templates according to the format specified by a client. It might include different normalisation operations for dates, time, currencies, etc. For instance, a round-off procedure for percentages can be executed [11, 14].

Not all of the tasks must be necessarily accomplished within one information extraction project. Therefore, a particular information extraction system does not have to have all of those possible components. According to Appelt and Israel [1] there are several factors that affect the choice of systems' components, like:

- Language. For processing texts in Chinese or Japanese languages with not clear word and sentence boundaries or texts in German language with words of a difficult morphological structure some modules are definitely necessary compared to working with English documents.
- Text genre and properties. In transcripts of informal speech spelling mistakes might occur in addition to implicit sentence boundaries. If information must be extracted from such texts those issues must be taken into account and addressed while designing a system by adding corresponding modules.
- Extraction task. For an easy task like names recognition the parsing and anaphora resolution modules might not be needed at all.

## 4. SOFTWARE ARCHITECTURES FOR INFORMATION EXTRACTION SYSTEMS DESIGN

In order to create an information extraction system the components which perform the stages mentioned above must be gathered into one pipeline. At the earliest stages of the development of information extraction as a field of study research groups designed information extraction systems from scratch every time they faced a different extraction problem. That was partly because at that time the major task was to solve the extraction problem and reusability of the tools created was not considered at all. Later, when the need for the integration of the tools developed by different groups was realised it was almost impossible to accomplish that task because of the diverse programming platforms used and the fact that the tools were not meant to be used in another application [12].

Since then several architectures have been developed to facilitate the process of the information systems development by providing the common platform for systems' components design, integration and reuse. Among them are the Unstructured Information Management Architecture (UIMA), the General Architecture for Text Engineering (GATE), the Architecture and Tools for Linguistic Analysis Systems (ATLAS), the Automated Linguistic Processing Environment (ALPE) [4]. Employing either of them it is possible to:

- Reuse the tools for natural language processing and text mining which have been previously created by other developers.

- Quickly combine different tools and thereby analyse possible approaches to design of the language processing software.

The first two architectures (UIMA and GATE) are the most prominent and provide almost the same capabilities.

UIMA was created by IBM and then became an Apache open-source project. Both Java and C++ frameworks are available. One of the major distinguishing features of UIMA is a *Common Analysis Structure* (CAS) which represents an original document and its stand-off annotations. Thus, the UIMA processing engine works as following. A *CAS Initialiser* acquires raw documents through the *Collection Reader* interface and produces the initial CASs. Then *Text Analysis Engines* (such as language translators, grammatical parsers or document classifiers) perform the document-level analysis, modify the CASs and transfer them to the *CAS Consumers*. The latter in their turn execute the collection-level analysis. It can be said that the main interface within the UIMA processing engine takes CASs as input and returns them as output [7].

GATE is an open-source architecture written in Java which was created by the University of Sheffield. One of the main elements of GATE is the *GATE Document Manager* (GDM). The GDM model includes three elements: a *collection* with *documents* which contain texts and *annotations* upon them. Thus, the GDM stores all the information about the texts which is produced by the system. All the components of the system interact with each other only through GDM which decreases the number of communication interfaces to one. CREOLE, a Collection of Reusable Objects for Language Engineering, is the GATE element which performs all the tasks of text analysis [3].

In the case of UIMA the unstructured data sources can be not only just plain text or HTML page, an audio or video streams can be processed as well. GATE in its turn supports XML, HTML, RTF, SML formats and plain texts [4]. Both GATE and UIMA have the graphical user interface for tools searching, browsing and integration. To upload an existing text analysis tool to the collection of predefined components existing within the both architectures a wrapping procedure must be performed. To be integrated into UIMA a tool must be written in C++, Java, Perl Python or TCL. The C/C++, Java, TCL, Prolog, Lisp and Perl tool's implementations are right for GATE [3, 12].

Thus, with the advent of such common frameworks as UIMA and GATE a huge step forward has been made in the development of the text mining technologies in general and in the information extraction area in particular. The latter has become more efficient since the researchers can draw on the other researchers' successful experience and have a platform for quick systems design.

## 5. TWO EXTRACTION APPROACHES

No matter which architecture is used to combine the components of the information extraction system it supports one of the two basic approaches of extraction, namely, *Knowledge Engineering Approach* and *Automatic Training Approach*.

**Knowledge engineering approach**. In order to extract information from available texts using a system which supports a knowledge engineering approach a set of extraction rules must be written manually. A person who creates such a type of system, or is responsible for writing those rules (knowledge engineer) must be an expert in the knowledge domain chosen for extraction. Apart from that, a designer must know the formalism for writing those rules for the particular system used. Usually the knowledge engineer has a number of texts which are related to the chosen domain. Analysing those texts the designer finds common patterns in them and writes the rules using his or her intuition, which according to Appelt and Israel [1] is a very important factor in creating a system with a high level of performance. The rules are then interpreted by the components of the information extraction system and useful facts are found and extracted from the texts. Creating an information extraction system using this approach is a highly time and effort consuming iterative process. Firstly, the knowledge engineer writes a particular rule. Then he applies it to the available texts and checks whether it works correctly or not.

Modifications are done if needed and the rule is examined again until a desirable result is achieved. Since this approach involves writing rules, in some sources it is called as a *rule-based approach*.

**Automatic training approach**. In this case there is no need to design extraction rules manually. Therefore a person who is responsible for the information extraction process does not have to know how to write rules and how a system works. A machine learning algorithm implemented in the information extraction system creates those rules. In order to do that the algorithm must have access to a large number of training texts related to the chosen domain. Those texts must be annotated manually in advance to provide examples on which the algorithm can learn and produce extraction rules. Thereby, the engineer must provide the set of training documents and be able to annotate them. Among algorithms that can be used for the automatic training approach there are decision trees, maximum entropy models and hidden Markov models [1]. In many sources this approach is named as the machine learning approach. The development of this method allows the information extraction area to become less domain-independent since the same machine learning algorithm can be applied to different domains as long as corpora of domain-related texts are available.

However, it is not necessary to create all the components of an information extraction system using only one particular approach. It is quite possible to interchange these two approaches while building different components of the system. One of the reasons of having such a possibility is that one can never say objectively which approach is better. Both of them have their advantages and disadvantages.

As Appelt and Israel [1] stated, the systems which use a knowledge engineering approach show a higher performance compared to the other ones. However, they require a lot of effort and time and depend on the knowledge engineer's skills and experience and availability of linguistic resources. The very important advantage of a machine learning based system is that it can be transferred to a different domain easily as long as specific texts and a person who can annotate them are available. But sometimes those texts are problematic or expensive to obtain or there is a lack of useful documents on which an algorithm can learn, and manual (or even machine-aided) annotation on the scale needed to provide reasonable levels of performance may be expensive.

On the basis of analysing the benefits and drawbacks of both approaches it is possible to conclude with the criteria which determine the choice of one of them. The most important condition to choose the automatic training approach is the presence of a set of suitable texts which can be used to train the algorithm. In the case of the knowledge engineering approach the availability of a person who is experienced in writing extraction rules is the most crucial criterion. Other aspects which can be considered are the specifications and the level of performance. If the specifications are subject to change and the level of performance is desired to be as higher as possible it is more reasonable to apply the rule-based approach, otherwise machine learning mechanisms can be employed.

## 6. INFORMATION EXTRACTION SYSTEMS EVALUATION

One of the ways to compare the extraction approaches is to measure the level of performance of the systems which support them. Message Understanding Conferences or Message Understanding Competitions (MUCs) have played an important role in the establishing of metrics to evaluate the information extraction systems. This conference was initiated by the United States Naval Ocean Systems Centre (NOSC) and was sponsored by the Defence Advanced Research Project Agency (DARPA). MUCs took place seven times from 1987 until 1998. Although the event is called a "conference", it can be described with other words like "competition" between information extraction research groups or "evaluation" of their systems' performances [2, 8, 14].

The major aim of these conferences was the evaluation of the state-of-the-art in the information extraction area, discovery and promotion of the new approaches in this field. However, Grishman and Sundheim (1996) claimed that MUCs differed from any other conferences in the way how the research groups were selected in order to take part in those conferences. The evaluation procedure started approximately 6 months before each conference. The research teams were given the same task to extract particular information from the sample texts. From year to year the tasks and domains were changed. The

research groups had to develop information extraction systems to accomplish those tasks. Just before the conference the participants were given a number of test texts to be processed using their systems. Then the obtained results from researchers were evaluated and compared with the pattern which had been gained previously by hand [8, 14].

The domain for the first conference was *Naval Tactical Operations*. Starting from MUC-2 different domains was explored, such as reports about *Joint Ventures*, *Terrorist Attacks* and *Airplane Crashes*. For the second conference, the same area of military messages as for the first one was chosen but the particular task identified by the organisers was to fill in a template with 10 slots for information to be extracted. From conference to conference, new tasks were introduced and they became more complex; the number of slots to be filled in increased constantly and texts in Japanese language were used alongside documents in English [14].

The MUCs showed that information extraction is not an easy task, as it is very difficult to create a system with an accuracy level of 100%. This means there is always relevant information in the text which is not extracted and extracted entities in the slots which are not relevant to the task. To evaluate information extraction processes the two metrics, namely, *Precision* and *Recall* were established. In simple terms, *Precision* (*P*) is the proportion of correctly extracted entities ($N_{correct}$) to the total number of extracted entities ($N_{response}$). *Recall* (*R*) is the proportion of correctly extracted entities ($N_{correct}$) to the total number of entities which are extracted manually ($N_{key}$). Thus,

$$P = \frac{N_{correct}}{N_{response}}, \qquad R = \frac{N_{correct}}{N_{key}}.$$

Another way of representation of information extraction systems evaluation is based on the notion of true and false positives and true and false negatives. It can be said that correctly extracted entities are true positives, whereas false positives are wrongly extracted information. Similarly, false negatives are relevant but not extracted information which is left in the text; true negatives are the information which is not extracted and not relevant to the task [1, 9]. Recall can be described as the measure of extraction effectiveness, whereas precision is the measure of extraction purity. Both of them are desired to be high. However, they are mutually dependent. If one of the metrics is increasing it leads to another metric decreasing and this trade-off is unavoidable.

In order to combine precision and recall, the *F* measure was introduced in one of the MUCs. Thus,

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \ 0 < \beta \leq 1.$$

It is the harmonic mean of two metrics which allows comparing and assessing different information extraction systems using one common base. Different values for $\beta$ are used to, e.g., favour precision over recall [1, 14].

The work that had been done through all the MUCs led to the formulation and introduction of basic extraction tasks. MUC-6 and MUC-7 contributed the most to this process. The Named Entity Recognition (NER) task is the first step of any information extraction system which involves proper names and quantities identification. The techniques used to accomplish this task are well-understood now and NER can be considered as a more or less "solved problem". The Template Element task (TE) is the next step to identify not only names but the descriptions of those names as well. The Template Relationship (TR) task implies finding the relationships between the entities extracted during the previous tasks. The Scenario Template (ST) task is based on the extraction according to the description of the particular event. The goal of the final Coreference task (CO) is to determine all the nouns, pronouns and noun phrases that refer to the same entity [8, 14].

After the last Message Understanding Conference 7 in 1998, the evaluation of information extraction systems has not stopped. MUC has been followed by the Automatic Content Extraction (ACE) programme since 1999. However, ACE is not just a copy of MUC; it differs from its predecessor in the following several ways [5]:

- ACE defined 3 main extraction tasks different from those of MUC. The tasks are: Entity Detection and Tracking, Relation Detection and Tracking, Event Detection and Characterisation. The first task involves the extraction not only of the name of an entity but anything that refers to that name, such as a description or a pronoun. That is why it is possible to say that the Entity Detection and Tracking task has combined the Named Entity Recognition and the Coreference Tasks of MUC.
- Not only English language texts, but texts in Arabic and Chinese are processed as well.
- Not only text documents but audio and image data are used to extract information from.
- Until 2008 the evaluation results had not been published. In 2008 the official results of ACE were made publicly available for the first time [15].
- The systems are evaluated using a *Value* measure which shows the correctly detected and recognised objects and their attributes. It is applied for all of the tasks and target objects, namely, entities, relations and events.

## 7. CONCLUSION

Information extraction is a relatively new area of study. However, as any information technology it advances quite quickly and a great progress has been made from the time it appeared. Texts from different domains were processed within the MUC and ACE competitions and the performance, for example, in the named entity recognition task has reached higher than 90% level. The process of the information extraction system design has changed from independent development of a system for a particular task from scratch to application of architectures like UIMA and GATE which allow using the previously created components and combining them easily.

However, there are still many unsolved problems. Event extraction task, for instance, cannot be executed as yet with as high level of performance as named entity recognition. And domain-independent information extraction systems are still one of the big research issues.

## REFERENCES


[1] Appelt D., Israel D. (1999) *Introduction to Information Extraction Technology*: IJCAI-99 tutorial <http://www.ai.sri.com/~appelt/ie-tutorial/IJCAI99.pdf> (Accessed on 10/04/11).

[2] Cowie J., Lehnert W. (1996) Information Extraction, *Communication of the ACM*, 39(1), pp. 80-91.

[3] Cunningham H. (2002) GATE, a General Architecture for Text Engineering, *Computers and Humanities*, 36(2), pp. 223-254.

[4] Dietl R., Hoisl B., Wild F., Richter B., Essl M., Doppler G. (2008) Project Deliverable Report. Deliverable D2.1 – Services Approach & Overview General Tools and Resources.

[5] Doddington G., Mitchell A., Przybocki M., Ramshaw L., Strassel S., Weischedel R. (2004) The Automatic Content Extraction (ACE) Programme – Tasks, Data and Evaluation, *Proceedings of the Conference on Language Resources and Evaluation*.

[6] Feldman R., Sanger J. (2007) *The Text Mining Handbook: Advanced Approaches In Analysing Unstructured Data*. New York: Cambridge University Press.

[7] Ferrucci D., Lally A. (2004) UIMA: an Architectural Approach to Unstructured Information Processing in the Corporate Research Environment, *Natural Language Engineering*, 10(3/4), pp. 327-348.

[8] Grishman R., Sundheim B. (1996) Message Understanding Conference – 6: A Brief History, *Proceedings of the 16th conference on Computational Linguistics*, 1, pp. 466-471.

[9] Grishman R. (1997) Information Extraction: Techniques and Challenges. In: Pazienza, M.T. (ed.) *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*. Berlin, Heidelberg: Springer-Verlag, pp. 10-27.

[10] Grishman R. (2003) Information Extraction. In: Mitkov, R. (ed.) *The Oxford Handbook of Computational Linguistics*. Oxford: Oxford University Press, pp. 545-559.

[11] Hobbs J.R. (1993) The Generic Information Extraction System, *Proceedings on the 5th Conference on Message Understanding*, pp. 87-91.

[12] Kano Y., Nguyen N., Sætre R., Yoshida K., Miyao Y., Tsuruoka Y., Matsubayashi Y., Ananiadou S., Tsujii J. (2008) Filling the Gaps between Tools and Users: A Tool Comparator, Using Protein-Protein Interaction as an Example, PSB 2008 Online Proceedings <http://psb.stanford.edu/psb-online/proceedings/psb08/kano.pdf> (Accessed on 10/04/11).

[13] Moens M.-F. (2006) *Information Extraction: Algorithms and Prospects in a Retrieval Context*. Springer Netherlands.

[14] Turmo J., Ageno A., Catala N. (2006) Adaptive Information Extraction, *ACM Computing Surveys*, 38(2), pp. 1-47.

[15] NIST 2008 Automatic Content Extraction Evaluation (ACE08). Official Results. Date of Release: September 29, 2008 <http://www.itl.nist.gov/iaui/894.01/tests/ace/2008/doc/ace08_eval_official_results_20080929.html> (Accessed on 10/04/11).