Machine Learning Approach for the Automatic Annotation of Events

Aymen Elkhlifi[1] and Rim Faiz[2]
[1]LARODEC, ISG de Tunis, 2000 Le Bardo, Tunisie.
aymen_elkhlifi@yahoo.fr,
[2]LARODEC, IHEC de Carthage, 2016 Carthage Présidence, Tunisie.
Rim.Faiz@ihec.rnu.tn

Abstract

After the beginning of the extension of current Web towards the semantics, the annotation starts to take a significant role, since it takes part to give the semantic aspect to the different types of documents.
With the proliferation of news articles from thousands of different sources now available on the Web, summarization of such information is becoming increasingly important.
We will define a methodological approach to extract the events from the news articles and to annotate them according to the principal events which they contain.
Considering the large number of news source (for examples, BBC, Reuters, CNN…), every day, thousands of articles are produced in the entire world concerning a given event. This is why we have to think to automate the process of annotation of such articles.

Introduction

The indexing of the documents and the extraction of events from them are increasingly becoming tiresome, since we are urged to generate an easily consultable semantic annotation to include or understand the increase in size of the document and to enrich its indexing.
By seeking an event given via a sequential course of the article, we meet sentences which do not refer to any event. Several other sentences refer to the same event. That's why we plan to eliminate the non event sentences and to group the others in the form of cluster by event.
Our research focuses on the annotation of document: first, we filter the non event sentences. Second, we group the sentences indicating the same event. Then, we generate the annotation.
The rest of the document is organized as follows: Section 2 introduces the related work on methods of annotation, then, the particulars methods of event annotation. In section 3, we present our task of automatic event annotation. In order to validate our survey, we describe the different progressive steps we followed to carry out the AnnotEvent system. The process of annotation is described in section 4. Section 5 states the evaluation of the system in order to demonstrate its capability. Section 6 concludes our work with a few notes on future work.

Related Work on Methods of Annotation

In their work, C. Roussey, S. Calabretto and J-M Pinon (2002) develop a tool of annotation for the semantic web called SyDoM. It processes the web page in XML format; it clarifies associated knowledge to a web page by the use of annotations and enables the multilingual research.
S. Tenier, A. Napoli, X. Polanco and Y.Toussaint (2006) developed an automatic WebPages semantic annotation system. The objective is to classify pages concerning teams of research, in order to be able to determine for example who works where, on what and with whom (use of ontology of the domain). It consists, first, of the identification of the syntactic structure characterizing the relevant element in the web pages, Then, of the identification of the most specific concept in the ontology in which the instance will be used to annotate this element. Their approach relies on a wrapper-based machine learning algorithm combined with reasoning making use of the formal structure of the ontology. However, in this approach, the exploitation of the arborescent structure of the page presents some limits according to the page regularity. It applies for documents of tabular type containing multiple instances of concepts of the ontology.
In their work, D. Brahim and al. (2006) developed an automatic engine called EXCOM for semantic annotations based on linguistic knowledge and making use of XML technologies. They are persuaded that using linguistic information (especially the semantic organization of texts) can help retrieving information faster and better on the web. The basic aim of this engine is to construct automatically semantic metadata for texts that would allow to search and extract data from texts annotated on that.
The work of J. Kahan and M-R. Koivunen (2001) belongs to the attempts of Semantic Web. In their system, the annotations are stored on waiters as metadata and are presented to the user by the means of a customer able to interact with the waiter by using protocol HTTP.

All preceding works are interested in the annotation of the documents like scientific articles, Web documents and multimedia documents. There exists others works which the web services (Abhijit and al, 2004). Only few works are interested in the annotation of the events. Among these works we can mention:

P. Muller and X. Tannier (2004) focused their work on the automated annotation of temporal information in texts, more specifically on relations between events introduced by verbs in finite clause. Both propose a procedure to achieve the task of annotation and a way of measuring the results. They have been testing the feasibility of this on newswire articles, with promising results. Then, they develop two measures of evaluation of the annotation: Fineness and Consistency.
In their work, A. Setzer and R. Gaizauskas (2000) present an annotation scheme for annotating features and relations in texts which enable to determine the relative order and, if possible, the absolute time of the events reported in them. Such a scheme could be used to construct an annotated corpus which would yield the benefits normally associated with the construction of such resources: a better understanding of the phenomena on concern, and a resource for the training and evaluation of adaptive algorithms to automatically identify features and relations of interest.
A.G. Wilson, B. Sundheim and L. Ferro (2001) present a set of guidelines for annotating time expressions with a canonicalized representation of the times they refer to, and describe methods for extracting such time expressions from multiple languages. The annotation process is decomposed into two steps: flagging a temporal expression in a document (based on the presence of specific lexical trigger words) and identifying the time value that the expression designates, or that the speaker intends for it to designate.

We note that work of annotation of temporal information generally concerns: detecting dates and temporal markers, detecting event descriptions and finding the date of events and the temporal relations between events in a text.
In our study, we are interested rather in the annotation of the events in the form of metadata on the document (we chose the news articles).

Approach of Event Annotation

Our approach of annotation of the events consists in:

- Extracting sentences comprising an event.
- Grouping those which refer to the same event in a cluster.
- Deducing the annotation in various forms.

The different steps of this process are as follows:

First step: Segmentation

In the first step some of the techniques of Natural Language Processing are applied to the texts in order to extract the sentences as well as the temporal markers which connect them (for details cf. to Faiz and Biskri, 2002, Faiz, 2006).

There are several systems which carry out this task:
The SegATex application (Automatic Segmentation of Texts), as a computer module, is intended to prepare (to tag) a text for an automatic language processing which includes text segmentation in sections, sub sections, paragraphs, sentences, titles and enumeration(SegATex, G. Mourad, 2001).
The "Lingua::EN::Sentence" module contains the function get_sentences, which splits text into its constituent sentences, based on a regular expression and a list of abbreviations (built in and given). Certain well-knowns. But some of them are already integrated into this code and are being taken care of. Still, if you see that there are words causing the get_sentences to fail, you can add those to the module, so it notices them.

While taking as a starting point these two systems, we have developed our own system SEG-SEN which splits up a given text into sentences while being based on the structure of the sentence and the punctuation marks.

Second step: Classification

During the second step, a model of classification is built automatically from training set which makes it possible to predict whether a sentence contains an event or not, due to the diversity of the techniques of machine learning.
We chose the decision tree for many reasons: It is easily interpretable by people. Also it's less skeletal compared to the other techniques which allow the reduction of the system's complexity.
We compare between the PCCs (Percentage of Correct Classification) resulting from various algorithms (of constructing of the decision tree).
Then, we choose the resulting data model which has the largest PCC. The result of this step is the sentences referring to an event.

In our study, we use the attributes which refer to the events. As defined by Naughton and Al (2006), these attributes are as follows: Length of the sentence, position of the sentence in the document, numbers of capital letters, numbers of stop words, number of city/town and number of numerical marks. We also added the attribute number of calendar terms.
The Training set is annotated by experts. For each news article, the events are annotated as follows:
The annotator is brought to assign labels for each sentence representing an event. If a sentence refers to an event, they assign the label "yes" if not "No ".

Third step: Clustering

We gather the sentences referring to the same event by the application of the algorithm ' Hierarchical Agglomerative Clustering (HAC) ' which initially assigns each object with a cluster, then collects on several occasions the clusters until one of the criteria of stop is satisfied (Manning and Schultz 1999).

HAC uses a measurement of similarity between the objects. For our case, we propose a new measurement of similarity between the sentences.

Similarity between sentences

Similarity measurement, in general, is based on the distance (Euclidean, Manhattan, Minkowski or that of Entropy), for the similarity between the sentences we find mainly the Cosines.

We can easily adopt the index of Jaccard for the sentences. If we replace the document by the sentence in his formula we get $S_{ij} = m_c / (m_i + m_j - m_c)$.

The index of similarity is the number of common words divided by the total number of words minus the number of common words:

m_c: Number of common words.

m_i: Size of the lexicon of the sentence S_i (i.e. number of different words in S_i).

m_j: Size of the lexicon of document S_j.

Measure of the Cosines

For the Measurement of the Cosines, we use the complete vectorial representation. Several methods to measure similarity exist; we quote the method based on the Finite State Automaton (FSA) developed by MDI (Thollard, Dupont and Higuera, 2000) and the method of TF-IDF Clustering Suggested by (Naughton, Kushmerick and Carthy 2006).

Finite State Automaton

Formally, let $L = \{l_1 l_2 ... l_n\}$ be a sequence of n event labels. We define:

$P(I(l_1))$ as fraction of the document that begins with the event label l_1. Similarly, $P(F(l_n))$ is the fraction of the document that ends with the event label l_n, and $P(l_{i+1} / l_i)$ is the fraction sentence labelled with l_i that are followed by sentences label with label l_{i+1}.

$P(L)$ is the probability that event sequence L of event is generated by the automaton.

$P(L)$ is defined as follows:

$$P(L) = P(I(l_1)) \times \prod_i P(l_{i+1} / l_i) \times P(F(l_n))$$

By using algorithm MDI (Thollard, Dupont and De la Higuera 2000) we train a Finite State Automaton from the sequences, where: The states correspond to the events labels and the transitions correspond to the adjacent sentences that mention the pair of events. The parameters of the automat are released by training on the document.

According to (Naughton, M and al 2006), let $L(c_1, c_2)$ be a sequence of labels induced by merging two clusters c_1 and c_2. If $P(L(c_1, c_2))$ is the probability that sequence $L(c_1, c_2)$ is accepted by the automaton, and let Cos (c_1, c_2) be the cosine distance between c_1 and c_2. We can measure the similarity between c_1 and c_2 as:

$SIM(c_1, c_2) = P(L(c_1, c_2)) \times Cos(c_1, c_2)$.

Iterative-TFIDF Clustering

Let's have S_1 and S_2 as sentences. The measurement of the similarity between S_1 and S_2 is defined as follows:

$$SIM(S_1, S_2) = \frac{\sum_{j=1}^{t} S_{1j} S_{2j}}{\sqrt{\sum_{j=1}^{t} S_{1j}^2 + \sum_{j=1}^{t} S_{2j}^2}}$$

With S_{ij} the weight of term t_i in the cluster j.

This weight is defined by (Naughton and al., 2006)

$W(t, c) = tf(t, c) \times ln\left(N / df(t_i)\right)$ with:

$tf(t, c)$: Frequency of the term t_i in the cluster c

N: Numbers of cluster.

$df(t_i)$: Cluster containing numbers the term l_i .

The first method (Finite State automaton) is too skeletal. The second method is effective but does not take into account the position of the sentence in the document. That's why it is syntactic. Indeed, it considers the word killed different from the word died which makes the similarity between the two sentences relating to both word weak.

We propose to extend this measurement in order to take into account the semantic aspect of the words and the position of the sentences in the article.

To be more semantic, we replace the words by their classes from ontology.

Examples:

Event1: *In Baquba, two separate shooting incidents Sunday afternoon left six dead and 15 wounded.*

Event2: *In other attacks reported by security and hospital officials, two car bombings in the northern city of Kirkuk killed 10 and wounded 32, and a blast in the southern city of Basra killed five and injured 15.*

We replace killed and dead by their class died. We replace also shooting incidents and bus bombings by their class explosion.

The semantic measurement of similarity between sentences becomes:

$$SIM(S_1, S_2) = \frac{\sum_{j=1}^{t} ct_{1j} ct_{2j}}{\sqrt{\sum_{j=1}^{t} ct_{1j}^2 + \sum_{j=1}^{t} ct_{2j}^2}}$$

It is important to group the sentences indicating the same events since they will be gathered even if they use various words.
We take into account in our function the position and the similarity between the sentences.

For the position, we express the position of a sentence in an article as follows:

$ct_i = \frac{Order\ (Sen)}{NbSen}$ with:

Order (Sen): Is the number of the sentences in the document.
NbSen: Is the total number of the sentences in the document.

This formula was used since the phase of classification to calculate the attribute position of the sentence in the document.
The distance between two sentences is defined by
Cos (ct_1,ct_2) with. $ct_i \in [0, 1]$

We propose the new measurement of similarity FSIM like a combination of the similarity between the sentences and the distance between them
FSIM (S_1, S_2) = $\alpha \times SIM(ct_1, ct_2) + (1- \alpha) \times \text{Cos}\ (ct_1, ct_2)$

Examples:
Applying algorithm HAC by using FSIM

C_1 Iraqi leader denies civil war as 50 people die.
C_2 On a day in which at least 50 people were killed, Iraqi Prime Minister Nuri al-Maliki said he did not foresee a civil war in Iraq and that violence in his country was abating.
In Iraq, we'll never be in civil war," al-Maliki told CNN's "Late Edition" on Sunday.
C_3 One U.S. soldier was killed by gunfire in eastern Baghdad about 2 p.m.
C_4 U.S. commander wounded since 1 p.m

The sentences in bold indicate an event. We calculate FSIM between these sentences.
Initially each sentence is a cluster we obtain these values:

FSIM (C_1, C_2) = 1.07
FSIM (C_1, C_3) = 0.12
FSIM (C_1, C_4) = 0.1
FSIM (C_2, C_3) = 0.08
FSIM (C_2, C_4) = 0.1
FSIM (C_3, C_4) = 0.06

Grouping together C_1 and C_2 into only one cluster C_A and recount FSIM for the new clusters:

FSIM (C_A, C_3) = 0.27
FSIM (C_A, C_4) = 0.21
FSIM (C_3, C_4) = 0.73

Grouping together C_3 and C_4 into only one cluster C_B and recount FSIM:
FSIM (C_A, C_B) = 0.14.

The process is stopped since 0.14 < 0.5. The threshold (0.5) is fixed like a stop criterion.
For N cluster there are *(n (n-1) / 2)* possible combinations.

Fourth step: Annotation

Using the clusters and their positions in the article, we generate a description which combines the events and which will constitute the annotation of the article under three types of annotations:

Sentence which annotates the cluster.
To structure the annotation in a standard form and possibly to store events in data bases.
To extract the concepts which represent the events in the article (future work).

First type of annotation: The sentence which annotates best the cluster is the one which contains the maximum values of the attributes used during the phase of classification. There is not much loss of information since the sentence which annotates the cluster is one among a set of similar sentences.
Such an annotation can be indexed to improve research of information on such articles, as it can be useful for an automatic abstract.

For the previous example we annotate the first cluster by:
Iraqi leader denies civil war as 50 people die.
The second cluster by:
One U.S. soldier was killed by gunfire in eastern Baghdad about 2 p.m.

Second type of annotation: To structure the annotation, we use the algorithm developed by Evens and al. (2004).

Example:
One U.S. soldier was killed by gunfire in eastern Baghdad about 2 p.m.
We will extract the following attributes:
Keyword: Killed
Location: Eastern Baghdad
Time/date: 2 p.m
Person: One U.S. soldier.
For each cluster we store these attributes in a data base that allows an easy research of the event by one of the attributes.

Experimentation

In our experiments, we employ a collection of 1210 news articles coming from 25 different sources describing the events related to Iraq war in 2006.
The news articles are of variable size (average length of the sentences by document is 18.75). The average of the events by document is 6.09.
After removing the images and the legends of the article, we segment them in to sentences by using the segmentator SEG-SEN which we developed in Java to extract the sentences on base of the structure of the sentence and punctuation markers.

Training set is part of the group of obtained sentences. It is annotated by two experts. For each sentence of the article, the default value of the attribute "Event" is ' No' (sentence not indicating an event), the commentator has to put ' Yes' if the sentence refers to an event. A file of format *APRR* (input of Weka) is generated automatically for each article, it will be useful like a source data for the algorithms of classification. We adopted J48, ADTREE and Random Tree with the cases of the events. We use *Weka 3.5*; because it allows us the access to source code to adopt it.

To evaluate the method of clustering, we employ the definition of the precision and the recall proposed by (Hess and Kushmerick, 2003). We assign each pair of sentences in one of the four following categories:

a: Grouped together (and annotated like referring to the same event).

b: Not grouped together (but annotated as referring to the same event).

c: Grouped inaccurately together.

d: Correctly not grouped together.

The Precision and the Recall prove that to be calculated as:

$$P = \frac{a}{a + c}, R = \frac{a}{a + b} \text{ and } F1 = \frac{2 \times P \times R}{(P + R)}$$

Results

The evaluation is done in several levels while starting with the evaluation of classification by using the PCC, then, the clustering by measuring the Precision and the Recall (Hess and Kushmerick 2003).

We exploited the following algorithms:
J48: implementation of C4.5 JR Quinlan (1993) which selected for each level the node of the tree as the attribute which differentiate better the data, then divided the training set in sub-groups in order to reflect the values of the attribute of the selected node. We repeated the same treatment for under group until we obtain under homogeneous groups (all the authorities or the majority have the same attribute of decision).

ADTree: construction of the decision trees extended to the cases of multiclasses and multi-labels.

Random Tree: begin with tree random and chosen by the majority best vote.

We obtained the following PPC:

J48

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.625	0.158	0.769	0.625	0.69	Yes
0.842	0.375	0.727	0.842	0.78	no

ADTREE

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.625	0.158	0.769	0.625	0.69	Yes
0.842	0.375	0.727	0.842	0.78	no

RandomTree

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.5	0.211	0.667	0.5	0.571	Yes
0.789	0.5	0.652	0.789	0.714	no

We obtained an improvement of Recall (R) and Precision (P) and the function F1
R = 85%, P=87% and F1=73.333%.

This improvement is made thanks to the semantic measurement of similarity which we developed. Indeed it detects the similarity between the sentences even if it contains different terms.
On the other hand, the filtering of non event forms a good input of phase of clustering.

Conclusion and Future Work

In this article, we describe and evaluate a method of semantic annotations for the news articles.
We initially developed a segmentator which splits up a text into sentences while basing it on the structure of the sentence and the punctuation markers.

We develop a model allowing the prediction whether a sentence is an event or not. Then, we compare the PCC resulting from various algorithms of construction of the decision trees.

In the third stage, we put forward a new measure of similarity between events which takes into account the same time the position of the sentences in the article, and the semantics used to describe the events.

This new measurement of similarity was used by algorithm HAC to group the sentence referring to the same event.
In the fourth step we generate the sentence which annotates the cluster in a better way. The whole sentences can play the role of summary article; in addition, the annotation can be used to enrich the indexing.

We are extending our work in several directions. First,
We plan to use other techniques of classification for the second step, like the SVM which is effective for the case of the two classes.
At the end, we think of the fusion of event by the adaptation of MCT proposed by (Smets 91).

References

Abhijit, A., Patil, S., Oundhakar, A., Sheth, K. 2004, Semantic web services: Meteor-s web service annotation framework. *In Proceedings of the 13th conference on World Wide Web*, 2004, New York, USA.

Brahim, D., Flores, J.G., Blais, A., Desclés, J.P., Gael, G., Jackiewicz, A., Le Priol, F., Leila, N.B., Sauzay, B. 2006. EXCOM: an automatic annotation engine for semantic information. *In FLAIRS 2006*, Melbourne, Florida.

Desmontils, E., and Jacquin, C. 2002. Annotations sur le Web: notes de lecture. *In AS CNRS Web Sémantique 2002*

Evens, M., Abuleil, S. 2004, Event extraction and classification for Arabic information Retrieval Systems. *In International Conference on Tools with Artificial Intelligence.*

Faïz, R. and Biskri, I. 2002, Hybrid approach for the assistance in the events extraction in great textual data bases. *Proc. of IEEE International Conference on Systems, Man and Cybernatics (IEEE SMC 2002)*, Tunisia.

Faiz, R. 2006. Identifying relevant sentences in news articles for event information extraction. *International Journal of Computer Processing of Oriental Languages,* World Scientific, Vol. 19, No. 1, pp. 19–37.

Kahan, J., Koivunen, M-R. 2001. Annotea: an open RDF infrastructure for shared Web annotations*, In Proceedings of the 10th international conference on World Wide Web.*

Naughton, M., Carthy, J., and Kushmerick, N. 2006. Clustering sentences for discovering events in news articles. *In Proc. European Conf. Information Retrieval.*

Muller, P., and Tannier, X. 2004. Annotating and measuring temporal relations in texts. 2004*, In Proceedings of Coling 2004*, volume I, pages 50-56, Genève, Association for Computational Linguistics.

Mourad, Gh. 2002. Analyse informatique des signes typographiques pour la segmentation de textes et l'extraction automatique de citations. Réalisation des Applications informatiques: SegATex et CitaRE, thèse de doctorat Université Paris-Sorbonne soutenance le 02 novembre 2001.

Mannig, C., and Schutze, H. 1999. Foundations of Statistical Natural Language Processing. MIT Press.

Mani, I., Ferro, L., Sundheim, B., Wilson. G. 2001. Guidelines for Annotating Temporal Information. *In Human Language Technology Confererence.* San Diego, California.

Naughton, M., Kushmerick, N., and Carthy J. 2006. Event extraction from heterogeneous news sources. *In Proc. Workshop Event Extraction and Synthesis*, American Nat. Conf. Artificial Intelligence.

Smets, Ph. 1991. The Transferable Belief Model and other Interpretations of Dempster-Shafer's Model. *Uncertainty in Artificial Intelligence 6*, P.P. Bonissone, M. Henrion, L.N. Kanal, J.F. Lemmer (Editors), Elsevier Science Publishers (1991) 375-383.

Quinlan, J. R. 1993. *Programs for Machine Learning.* Morgan Kaufmann Publishers.

Radev, D. R. 2000. A Common Theory of Information Fusion from Multiple Text Sources Step One: Cross-Document Structure. *In Proceedings, 1st ACL SIGDIAL*

Roussy C., Calabretto S., Ponon J.M. 2002, SyDoM : un outil d'annotation pour le Web sémantique. *In Proceedings of Journées Scientifiques Web sémantique.*

Setzer, A., Gaizauskas, R. 2000. TimeML: Robust specification of event and temporal expressions in text. *In The second international conference on language resources and evaluation.*

Thollard, F., Dupont, P., and De la Higuera, C. 2000. Probabilistic dfa inference using kullback-leibler divergence and minimality. *In Proceedings of the Seventeenth International Conference on Machine Learning.*

Tenier, S., Napoli, A., Polanco, X., and Toussaint, Y. 2006. Role instantiation for semantic annotation. *In International Conference on Web Intelligence, Hong-Kong, IEEE / WIC / ACM.*

Zha, H. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. *In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.113–120.