



Pattern Extraction for Event Recognition in the Reports of Polish Stockholders

Michał Marcińczuk and Maciej Piasecki

Institute of Applied Informatics, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, Wrocław, Poland, maciej.piasecki@pwr.wroc.pl

Abstract. In the paper the application of the general Memory Base Learning to Event Recognition in the domain of reports of stock issuers is investigated. A multi-classifier scheme is applied in which the boundaries of annotations are identified first and then a heuristic algorithm of merging into pair is applied. A modified method based only on positive examples is proposed. Several types of simple features requesting only simple processing of text are tested. The proposed method can be trained on a small annotated corpus.

1 Introduction

Event Recognition (ER) is a specific case of Named Entity Recognition and relies on detecting given type of events and their attributes. ER has been used to retrieve informations about terrorist activities from news (MUC-3, MUC-4), air plane crashes and rocket/missile launches (MUC-7) and medical data from mammographical reports [6].

The approaches to ER can be divided into three groups: manual approaches, automatic approaches and combinations of these two. The manual approach makes use of grammar [6], regular expressions [13] and predicate-argument models [14]. In the automatic approach the machine learning methods (decision trees [2]) and statistical models (Hidden Markov Model [15]) have been applied.

The manual approach allows to achieve very good quality (in terms of precision and recall) in comparison to the automatic approach. However, it has some disadvantages, like high cost in terms of time and human work to create the extraction patterns for a given type of annotation.

2 Event Recognition

ER is one of the Information Extraction tasks. It relies on detecting descriptions of events in text and extracting events attributes from text. An event is a fact that occurred in a time [1], for instance, the annual meeting of the stockholders announcement. Each type of event is characterized by set of attributes. For instance, the annual meeting of the stockholders announcement contains following information: the date, the hour and the place of the meeting, the meeting agenda, the date, the hour and the place of the trust deposit.

ER is a specific case of Named Entity Recognition [10, 5]. The core of Named Entity Recognition is finding descriptions of entities of the defined types in text. The task of ER is finding instances but also ER techniques take also the role of the instance into account. For example, the annual meeting of stockholders can be associated with two instances of the date, time and place but the instances have different roles. One refers to the meeting and the others refer to the trust deposit.

The goal of ER is to detect an event description in a text document and extract the values of the event attributes. Next, the found beginnings and endings of description are marked (annotated) with some *tags*.

The goal of this work is to construct a method of recognition of selected events in the domain of activities of Polish stock issuers. The method should be possible to be applied on the basis of a limited corpus annotated manually.

2.1 Event Recognition as Classification Task

In the naive approach ER can be treated as a single-classification task that it is performed by testing every subsequence of tokens, where a *token* is a basic segment of text e.g. a word form, number, date or some symbol in a document. However, in this approach some problems occur. The main problem is the number of instances to be tested. This is caused by variable lengths of annotations that are depending on the complexity of event descriptions. All different subsequences should be checked, i.e. the huge number of subsequences can be generated even for a sentence consisting of 20 tokens. The next problem is how to represent the problem by the means of a constant number of learning / testing features as most of the classification algorithms assume.

Fortunately, ER can be also considered as a multi-classification task. The idea is to decompose ER into several classification tasks and then merge the partial results into whole annotations. Bennett et al. [2] used two classifiers to recognize the beginnings and the endings of annotations. Pradhan et al. [11] used *IOB representation* (abbreviation stands for *Inside Outside Begin*) and created three classifiers that divide tokens into three classes: tokens beginning some annotation, being part of some annotation or being outside of any annotation.

Both approaches has the same problem that the non-boundaries tokens are strongly prevalent. This means that for every single positive instance there are hundreds of even thousands negative instances.

2.2 Instance Features

Descriptions of events are natural language expressions. The beginning and ending of expression can be characterised by several linguistic features that are used during training and classification. The groups of features presented below are put in order of increasing demands on the complexity of processing:

- *Rough characteristics of token characters*—type of characters occurring in token structure. There are 7 types of tokens distinguished according to types of characters constituting them. The types are presented in Table 1.

Table 1. Groups of tokens according to the characteristics of characters

Group	Description
LowerCase	a sequence of lower case letters
UpperCase	a sequence of upper case letters
InitCap	a capitalized word
MixedCase	a sequence of lower and upper case letters
Number	a sequence of digits
Symbol	a sequence of characters other than letters and digits
Mixed	a sequence of letters, digits and/or symbols

- *Morpho-syntactic description* of a word form (called also a morpho-syntactic tag)—we assume the IPI PAN Corpus (IPIC) format [12], according to which a description consists of: a *grammatical class* (more fine grained division of Parts of Speech into 32 classes), and value of grammatical categories, e.g. number, case, gender, person, degree and other. There are 12 grammatical categories assign in different subsets to different grammatical classes, see [12]. The morphological base form which is a part of an IPIC tag, is separated here because of its importance in the algorithms proposed here.
- *Morphological base form*—represents a set of word forms differing in values of grammatical categories but described as one lexeme in a lexicon. As assignment of morpho-syntactic tags to word forms is often ambiguous (many tags possible for a word form) for these two features the usage of tagger disambiguating description is necessary.
- *Shallow/deep syntactic structure*—is necessary in the construction of the *predicate-argument model* of a sentence that is the basis for the assignment of semantic roles to the sentence elements in [14].
- *Semantic groups*—defined on the basis of automatically extracted semantic similarity (semantic relatedness), or defined manually in the form of sets of synonyms, e.g. [7], could be used to divide word forms into semantic classes, that would simplify the classifier training and work. However, any such semantic resource has not been constructed for the domain of our interest, yet.

3 Memory Based Learning

Most of the Machine Learning algorithms needs a large set of training examples in order to achieve good results. In our domain of interest, the preparation of such set is very laborious as it means that hundreds of annotations must be manually introduced to text documents. Thus we assumed the general scheme of the *Memory Based Learning* as the basis for our solution. Decreased demands on the preparation of training data makes the proposed solution closer to practical applications.

In our approach, two separate classifiers are built: one for annotation beginnings, and the second one for endings. The identified beginnings and endings are next combined by a heuristic algorithm, see Sec. 3.2.

Both classifier use three features types (token type, base form and grammatical class) of three tokens before a position being classified (an *outer context*) and two tokens after (an *inner context*).

Base forms and grammatical classes are identified by the previous application of the TaKIPI tagger [9]. As two additional features, we introduced the first preceding noun and preposition. A sample training instance of date annotation is shown in Table 2.

Table 2. A sample training instance of date annotation starting boundary

feature no.		w	dniu	19	czerwca	2006
1-5	type	LowerCase	LowerCase	Number	LowerCase	Number
6-10	base form	w	dzień	19	czerwiec	1006
11-15	grammatical class	prep	subst	ign	subst	ign
16	preceding noun base	odbyć				
17	preceding preposition	w				

3.1 Original Approach

Memory Based Learning [3] (also known as *Instance Based Learning* or *Lazy Learning* [8]) relies on storing the training instances as they are. No generalization or reduction is performed at the training step so no information is lost. The problem of generalization is postponed until new instances are classified.

As all the features used in our experiments are symbolic, we have chosen the overlap metric (Equation 1 and 2).

$$\Delta(X, Y) = \sum_{i=1}^n \delta(x_i, y_i) \quad (1)$$

$$\delta(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Memory Base Learning applies the k-NN (k-Nearest Neighbours) method to classify new objects during working phase. The method assign this class to a new object that is most common among a group of the k most similar stored, training examples. A visualization of this method is presented on the Figure 1 on the left side., where triangle represent our tested object (i.e. a vector of feature values describing a given position in text), the circles negative training examples and the discs positive examples. The gray disc encapsulate the $k=5$ neighbors, which are nearest to the tested object. In this case the new object will be classified as a positive one.

3.2 Modified MBL

Because of the fact that negative instances (non-boundary tokens) are strongly prevalent we introduced a modification of the original Memory Based Learning methods that can cope with this problem. In the modified MBL only positive examples, which represent starting or ending boundaries, are stored. The negative examples (non-boundary tokens) are not taken into consideration at all. However, in that way for almost any position in text we can expect to have some positive neighbours in a distance greater than zero.

Because only examples of one class are stored, the k-NN methods cannot be used. Instead, we introduced a *threshold* for the number of neighbours written as the parameter k . The classification process starts with measuring the distance between a new object and all stored examples. Next, the number of examples whose similarity is greater or equal to the *value threshold* t is counted—only these are treated as neighbours in the next step. A new object is classified as positive only if the number of found neighbours is greater or equal to the parameter k . A visualization of this method is presented in Figure 1 (right side).

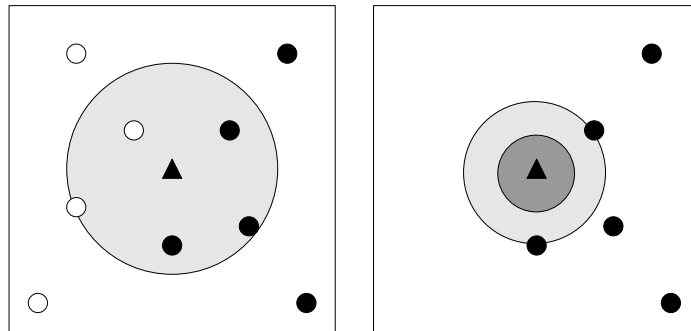


Fig. 1. Visualisation of the k-NN method (on the left) and the k -threshold classification (Modified MBL)(on the right)

After identification of possible beginnings and endings we have to identify pairs of them defining borders of subsequent annotations. The important domain factor is that the annotations do not cross, i.e. the beginning of the next one cannot be located between the beginning and the end of the previous one. This fact helped in constructing a heuristic procedure of finding pairs.

First, the highest values of similarity which were found during classification are stored for each classified object. Next sequences (possibly of the length one) of token classified as possible beginnings are identified in text. Then, inside each sequence the hill-climbing algorithm is used to find the token with the highest similarity. For the selected token, in the next step we are looking to the right for a token that can be an ending boundary. When we search for the ending boundary we consider only tokens that appear on the positions from the *starting boundary*

position + a *minimum length of the annotation to the ending boundary position* + a *maximum length of the annotation*. The values of the *minimum length of the annotation* and the *maximum length of the annotation* are calculated during the learning process.

4 Experiment

4.1 Training Set

For the needs of experiments, a set of training examples was extracted from documents concerning stock exchange domain. Each document contains a report posted by a stock issuer. In Poland every unit that issues stocks is obligated by law to present current and periodical information about the issuer according to act [4]. The act defines 26 types of information that the issuer is obligated to publish. During the several months of the year 2006 we collected a set of documents including more than 10000 documents. At the first step the documents were filtered in order to separate those describing the annual meeting of stockholders. 390 documents were found by simple keyword searching. Then each document was manually annotated in terms of the *date*, the *time* and the *place* of a meeting.

Finally, the training set consisted of **424** instances of the meeting date, **383** instances of the meeting time and **385** instances of the meeting place.

4.2 Base Line

Because it was hard to find a exactly similar work (especially for Polish), as a base line we chose the performance of the manually created regular expressions. We wrote a set of rules to extract each type of annotations (time, date and place) from plain text. A sample regular expression for meeting place:

```

1 (w|we)\s*
2 (?<annotation>
3   (siedzibie\s*[sS]polki\s*((,\s*)?w\s*)?)?\w*(,|\s*przy)\s*
4   (?<street>ul\.\s*\w*\s*(nr\s*)?(?<number>\d*(/\d*)?))
5 )

```

Table 3. Performance of the regular expressions

	Precision	Recall	F-measure
Meeting time	93.75	97.91	95.78
Meeting date	59.57	96.76	73.75
Meeting place	17.91	16.10	19.96

The construction of the regular expression took several working days. The results of the their application are presented in Table 3.

In the case of relatively simple patterns of time expression, the result is quite good. The outstanding low results for the *meeting place* annotation were caused by the two facts. First, for each type of annotation a 2-hours time limit was set on the work spent on the manual construction of the expression. The *meeting place* annotation is a combination of smaller elements like a city name, a postal code, a street name and number, a building name and others. The variety of the annotation elements (the order and the number) caused that it was difficult to create a good regular expression during the 2-hours limit time.

In the base line we consider only the performance for whole annotation recognition, we did not constructed and tested expressions for the identification of beginnings and endings, separately.

Table 4. Number of positive and negative instances for the meeting date annotation

	Positives	Negatives	Total
Balanced	424	424	848
1%	424	3015	3439
10%	424	30157	30581
100%	424	301579	302003

4.3 MBL Approach

Applying the original MBL, we have conducted 3 experiments with different number of negatives instances used in the training set. Each case was testes using 10-fold Cross Validation and the minimum, average and maximum vales of precision, recall and F-measure are presented. In the first run we used balanced number of positives and negatives instances. For each positive instance one randomly taken negative instance was used. In the following runs we used 1%, 10% and 100% of all negative instances.

In the original MBL approach we consider only the performance for starting and ending boundaries recognition, see Table 5 and Figure 2 (MBL precision, recall and F-measure), as the first results of the application of the heuristic search methods were not encouraging.

4.4 Modified MBL

In the modified MBL approach we calculated the performance for both starting and ending boundaries, as well, as whole annotation recognition (Tables 6, 7, 8).

Table 5. Original MBL with the $k=3$, all positive and 1% of all negative examples

	Begin			End			Annotation		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Min.	10,42	96,08	18,87	9,79	96,43	17,83	18,75	39,22	25,68
Avg.	14,78	98,84	25,61	13,37	99,31	23,52	22,24	46,30	29,92
Max.	20,33	100,00	33,56	16,09	100,00	27,72	26,67	54,90	35,90

Table 6. Results for the *meeting time* using modified MBL with the *threshold*=0.75

	Begin			End			Annotation		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Min.	84,444	90,244	90,476	81,395	87,500	84,337	97,143	85,000	90,667
Avg.	92,789	97,041	94,802	85,499	91,575	88,402	99,714	89,378	94,240
Max.	97,619	100,000	97,619	92,857	97,674	94,382	100,000	94,872	97,368

Table 7. Results for the *meeting date* using modified MBL with the *threshold*=0.75

	Begin			End			Annotation		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Min.	86,667	83,673	85,417	81,356	91,667	86,275	92,593	80,392	88,636
Avg.	92,167	88,269	90,116	88,085	95,106	91,408	98,332	85,184	91,230
Max.	97,826	93,333	92,784	96,667	98,039	96,667	100,000	90,000	94,737

Table 8. Results for the *meeting place* using modified MBL with the *threshold*=0.75

	Begin			End			Annotation		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Min.	40,000	85,000	55,474	51,220	45,238	50,602	44,828	29,268	36,923
Avg.	50,549	90,636	64,725	74,487	60,505	65,919	62,971	40,036	48,580
Max.	57,813	97,561	70,476	96,296	78,571	78,873	81,481	53,659	64,706

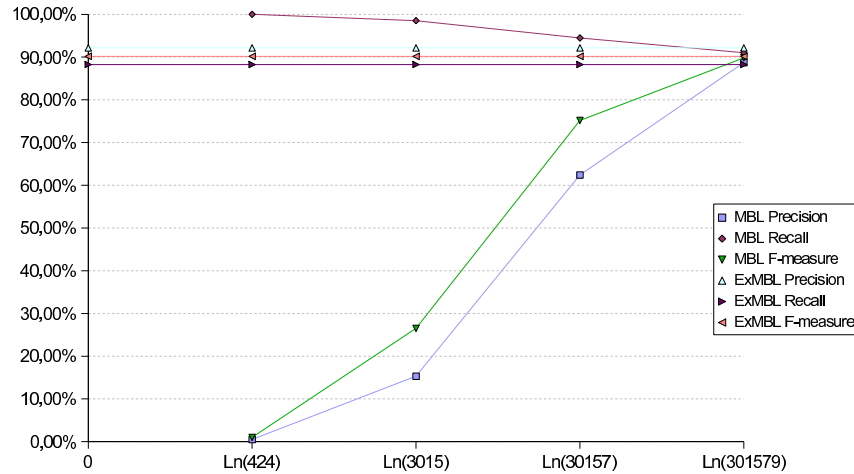


Fig. 2. The impact of the negative instances

5 Conclusions and Future Work

Event Recognition is a specific case of Named Entity Recognition and relies on recognizing events and their attributes. Event Recognition can be treated as a single classification problem as well as multi-classification problem. In both cases the negative instances (non-boundary tokens) are strongly prevalent. The experiment shown that the best results using Memory Based Learning were obtained when all instances (positives and negatives) were used in the training process. Reduction of the negative instances caused the decrease of the performance.

We proposed a simple modification of the original Memory Based Learning methods that uses only of the positive instances. The obtained results for the modified method are comparable to the original MBL results but only when in MBL all examples are used (Figure 2). However, when we decrease the number of negative examples, the MBL methods stops working in the case of this particular tasks and the domain. Contrary to this, the Modified MBL method produces good results for the number of training data was reduced about 1000 times. It means that time of processing is reduced enormously: the modified MBL works on fly, while the application of MBL for the whole training corpus takes days.

The experiments showed that for this particular task, a simple memory based learning can be a solution competitive to manually constructed rules, as a limited corpus is required which can be annotated in short time, the annotation of documents is much easier than the construction of rules, and the method works efficiently.

As we have only the TaKIPI tagger in our disposal, we wanted to stay with as simple features as possible. However, the natural direction of the further research is enrichment of the description of text structure, and adding semantic properties

of word forms. We want also to investigate different similarity functions, e.g. applying some domain heuristics.

References

1. The "Message Understanding Conference (MUC)" web page
http://www-nlpir.nist.gov/related_projects/muc
2. Scott W. Bennett, Chinatsu Aone, and Craig Lovell. Learning to tag multilingual texts through observation. *Proceeding of the Second Conference on Empirical Methods in NLP*, page 8, 1997.
3. Walter Daelemans and Antal van den Bosch. *Memory-Based Language Processing*. Cambridge University Press, 2005.
4. Dz.U.05.209.1744. *Rozporządzenie ministra finansów z dnia 19 października 2005 r w sprawie informacji bieżących i okresowych przekazywanych przez emitentów papierów wartościowych, dziennik ustaw z 2005 r. nr 209 poz. 1744*, <http://www.abc.com.pl/serwis/du/2005/1744.htm>
5. Xiaoshan Fang and Huanye Sheng. *Advances in Information Systems: Second International Conference, ADVIS 2002, Izmir, Turkey, October 23-25, 2002. Proceedings*, chapter Pattern Acquisition for Chinese Named Entity Recognition: A Supervised Learning Approach, pages 166–175. Springer Berlin / Heidelberg, 2002.
6. A. Kupść., A. Marciniak, A. Mykowiecka, J. Piskorski, and T. Podsiady-Marczykowska. Information extraction from mammographic reports. In *KONVENS 2004, Osterreichische Gesellschaft für Artificial Intelligence*, pages 113–116, 2004.
7. Derwojedowa Magdalena, Piasecki Maciej, Szpakowicz Stanisław, and Zawisławska Magdalena. Polish wordnet on a shoestring. In W G. Rehm, A. Witt, and L. Lemnitzer, editors, *Proceedings of Biannual Conference of the Society for Computational Linguistics and Language Technology*, pages 169–178. Universität Tübingen, 2007.
8. Tom M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997.
9. Maciej Piasecki and Grzegorz Godlewski. Effective architecture of the polish tagger.
10. Jakub Piskorski. *Intelligent Media Technology for Communicative Intelligence*, chapter Named-Entity Recognition for Polish with SProUT, pages 122–133. Springer Berlin / Heidelberg, 2005.
11. Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin, and Daniel Jurafsky. Semantic role chunking combining complementary syntactic views. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)*, pages 217–220, 2005.
12. Adam Przepiórkowski. *The IPI PAN Corpus: Preliminary version*. Institute of Computer Science, Polish Academy of Sciences, Warsaw, 2004.
13. Ellen Riloff. Automatically generating extraction patterns from untagged text. *Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, page 1044–1049, 1996. Portland, OR.
14. Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. *An improved extraction pattern representation model for automatic ie pattern acquisition*, *CiteSeer*, page 8, 2003.
15. Gökhan Tür. *A Statistical Information Extraction System For Turkish*. PhD thesis, Bilkent University, 2000.