Information Extraction with RapidMiner

Felix Jungermann

Artificial Intelligence Group, TU Dortmund, `http://www-ai.cs.tu-dortmund.de`

Abstract. In this paper we present the Information Extraction (IE)-plugin for the open source Data Mining (DM) software *RapidMiner*[1] [Mierswa et al., 2006]. The IE-plugin can be seen as an interface between natural language and IE- or DM-methods, because it converts documents containing natural language texts into machine-readable form in order to extract interesting information like special entities and relations between those. The plugin is very modular and easy to use, which makes it more applicable for different domains and tasks.

Key words: Natural Language Processing, Information Extraction, Information Extraction System, Named Entity Recognition, Relation Extraction, Structured Methods

1 Introduction

Nowadays more and more information is available spread all over the internet. The information is present on websites – containing pure text on the one hand and html-code on the other hand –, in documents (pdf-documents for instance), or in log-files and so on. To process this daily growing huge amount of information manually is impossible.
Therefore IE-techniques are used for *the automatic identification of selected types of entities, relations, or events in free text*, as [Grishman, 2003] says.
While some IE-systems process IE-tasks like for instance Named Entity Recognition (NER) in a somehow black-boxed way, we present a very modular system, which can easily be adjusted and extended for already known or new tasks.
In the following section we give a short overview of the history of IE. In section 3 we present the state-of-the-art DM software *RapidMiner*. In section 4 the special aspects of natural language processing and how they are respected in *RapidMiner* will be explained. Some principles concerning automatic IE are defined in section 5. The IE-plugin and an exemplary text-handling task will be presented in section 6. Finally section 7 will give a conclusion about current efforts and future work.

2 Information Extraction

Beside the definition of [Grishman, 2003], [Cardie, 1997] defines IE as a kind of summarization of texts according to some (predefined) topic or domain. Both

[1] http://www.Rapid-I.com

definitions aim at a deeper text understanding.
But before one gets to know how to gain more knowledge from unknown texts, one should have a look at the history of IE.

The field of IE was born together with the Message Understanding Conference (MUC) [Grishman and Sundheim, 1996] which were scientific, shared tasks, offering texts of a special domain. Participants in these tasks had to answer specific questions concerning several special mentions in the text. One of the first domains for example was 'terrorist acts'. The questions to answer were 'Who was the actor?', 'Who was the victim?', 'What happened?', and so on.
Answering these questions automatically is a non-trivial task. Because of that, many systems were handcrafted and contained a huge amount of manually designed rules, resulting in systems biased towards a specific domain.
Using these systems for other domains was impossible or very unattractive, because much human effort had to be invested to tune all the rules manually.
The understanding of the semantic of texts was not very deep, and therefore tasks were defined that need to be processed in every domain for a deeper understanding: namely *co-reference analysis*, *word sense disambiguation* and *predicate-argument structure*.
Especially a new task of the sixth MUC – namely NER – got very popular. In its original form, the task was to extract predefined (very basic) named entities like person names, locations and organizations.
The goal to create a new task was successful, but NER is in no way domain-independent, because every special domain needs special entities to be extracted.
A very popular application area of NER is the biological domain which uses NER for identifying genes, proteins and so on. That task is called 'bio-entity recognition', like in the BioNLP of 2004 ([Kim et al., 2004]).

Those entities are the first semantic milestones on the way to a better, automatic text understanding.
But, effective – domain- and language-independent – systems should not be based on handcrafted linguistic rules. Although manually tuned systems are sometimes better in terms of precision and recall (see section 5.3), machine-learning or statistical methods are used, because they are more flexible and do not rely on heavy human efforts.
Machine-learning systems used for NER are hidden Markov models [Rabiner, 1989], maximum entropy Markov models [McCallum et al., 2000], conditional random fields [Lafferty et al., 2001] and structured support vector machines [Tsochantaridis et al., 2005].

3 RapidMiner

In this section we will give a short overview about the concepts of traditional DM. The opensource DM-software *RapidMiner* – one of the best DM tools [2] –

[2] according to the `www.KDnuggets.com` poll 2007

is presented as an exemplary environment for DM-tasks.

One can say that Knowledge Discovery in Databases (KDD) is the whole field of research of extracting new and interesting knowledge out of (masses of) information, which in addition might be very unstructured. The term KDD is often used equivalently in one context with the term of DM, which describes the techniques used to extract that knowledge. Because the amount of information is often huge, the techniques must be scalable to work efficiently.
Traditionally this extraction was done on databases containing related entities. These relations are decomposed and put into a flat form in order just to have one table to learn from.
As it is the same convention in *RapidMiner* from now on this table is called exampleset, and this exampleset, again, contains examples. In addition, examples consist of attributes. The special *label*-attribute is the one to predict from unlabeled new examples. Table 1 shows such an exampleset. In this exampleset the label is a binary one.

Table 1. Possible exampleset in *RapidMiner*

$Example$	$Attribute_1$	$Attribute_2$	...	$Attribute_m$	$Label$
...	...	...	...	...	...
$example_{n-1}$	0.1	true	...	'B'	1
$example_n$	0.2	false	...	'Re'	-1
$example_{n+1}$	0.2	true	...	'Bc'	1
$example_{n+2}$	0.05	true	...	'BB'	1
...	...	...	...	...	...

3.1 Process-view

The process-view (see Figure 1) of *RapidMiner* offers a modular and pipelined view on the processed experiment. The process normally consists of four stages:

- the input-stage,
- the preprocessing-stage,
- the learning-stage,
- and the evaluation-stage.

These stages correspond to the CRISP-model ([Chapman et al., 1999]) which is a well-known standard for DM processes.

4 Information Extraction aspects

Remembering section 2, one sees that IE can be seen as a special case of traditional data mining.

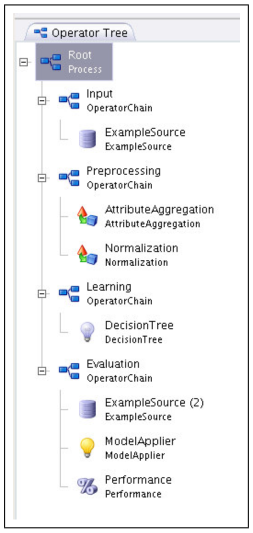


Fig. 1. The *RapidMiner* process-view

Examples will contain the tokens of the text , whereas the whole document(-collection) can be seen as the exampleset. The attributes are syntactic or semantic pieces of the tokens, and the label is the special semantic information one likes to predict.
One of the most important aspects in IE is the sequential character of texts. For the categorization of documents it is often sufficient to just use the so called *bag of words*, which is a simple index list of the contained words, to classify an unknown document – having seen enough classified documents during the training phase. But for extracting relevant information, IE needs the words itself and its contexts, e.g. to predict if it is an entity or not.
[McDonald, 1996] describes external and internal evidence as necessary to extract the correct semantics for a given word. The fact that *RapidMiner* is a DM software is remarkable because in 'traditional' DM a special example (and its attributes) originally is independent of the other examples – although they might likely share patterns. For IE all documents have to be converted to proper examples first. During conversion it is necessary to maintain the sequential character of documents and especially of sentences. Standard ML operators like sampling have to be adapted to meet the needs of the plugin in order to not destroy the structural information of the language constructs.
One way to maintain such structural characteristics is to associate words and sentences with identifiers, usually by using counter variables. Table 2 shows a possible exampleset for IE-tasks. For a better understanding the reader should have a look at section 6 which shows an example process for IE.

Table 2. Possible IE-exampleset in *RapidMiner*

$Token$	$Doc.No.$	$Sent.No.$	$TokenNo.$	$Attribute_1$	$Attribute_2$	...	$Attribute_m$	$Label$
...	...	...	...	...	...	...	...	...
goes	2	4	2	'g'	's'	...	aaaa	O
to	2	4	3	't'	'o'	...	aa	O
Hamburg	2	4	4	'H'	'g'	...	Aaaaaaa	LOC
because	2	4	5	'b'	'e'	...	aaaaaaa	O
...	...	...	...	...	...	...	...	...

5 Automatic-Information Extraction

In this section we explain two of the most important tasks in IE and their techniques that are present in the IE-plugin.

5.1 Named Entity Recognition (NER)

Traditionally, NER is the task to predict the best label-sequence Y for a given observation-sequence X, having seen a sufficient amount of training-pairs

$$< x^{(i)}; y^{(i)} >$$

There are some unsupervised approaches for NER (which do not need masses of training-data) ([Nadeau et al., 2006], [Collins and Singer, 1999], [Etzioni et al., 2005]) but they heavily rely on seed-examples and knowledge about the wanted Named Entity (NE)s.
The current state-of-the-art technique for NER are CRF we will explain now in more detail.

Conditional Random Fields (CRF) CRFs are undirected graphical models and have first been proposed by [Lafferty et al., 2001].

Definition 1. *Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a CRF, when conditioned on X, the random variables Y_v obey the Markov property with respect to the graph:*

$$p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$$

where $w \sim v$ means that w and v are neighbors in G.

The generated label-sequence is calculated on the whole graph conditioned on the complete observation-sequence. The probability of the graph is defined via two kinds of features:

- transition-features: $f(e, y|_e, x)$ defined on the edges of the graph and
- state-features: $g(v, y|_v, x)$ defined on the vertices.

In the simplest and most important example for modeling sequences, G is a simple chain or line:

$$G = (V = \{1, 2, ..., m\}, E = \{(i, i+1)\})$$

If the graph is a tree one can write the conditional distribution of Y and X as follows:

$$p_\theta(y|x) \propto exp\left(\sum_{e \in E, k} \{\lambda_k f_k(e, y|_e, x)\} + \sum_{v \in V, k} \{\mu_k g_k(v, y|_v, x)\}\right)$$

Using the log-likelihood we can derive the following optimization function $\mathcal{L}(\lambda)$ as:

$$\mathcal{L}(\lambda) = \sum_i \left(log\ \frac{1}{Z(x^{(i)})} + \sum_k \lambda_k F_k(y^{(i)}, x^{(i)})\right)$$

The goal is to optimize λ which is the weight-vector for the CRF-features such that $\mathcal{L}(\lambda)$ is maximized. This optimization can be done by using quasi-newton optimization (e.g. L-BFGS, [Liu and Nocedal, 1989]).

5.2 Relation Extraction (RE)

If the task of finding the entities is sufficiently done, one can look for relations between these entities. The scientific field of finding relations between entities has become popular since the ACE-tasks[3]. Especially the relation detection and classification task in 2004 [LDC, 2004] has heavily been worked on. If all entities in a sentence have been found, every possible pair of two entities is combined to a relation-candidate in order to find out wether there is a relation and to predict the corresponding relation type.

Composite Kernel The state-of-the-art techniques in this field of research are treekernel-methods based on the research done by [Haussler, 1999] and [Collins and Duffy, 2001]. A recent extension is the combination of treekernels on the one hand with linear kernels on the other hand, resulting in a so called composite kernel. The composite kernels have shown to achieve better results than with just one of these kernels ([Zhang et al., 2006], [Zhang et al., 2007]). These techniques are heavily based on structural information (parse-trees). Therefore the flat data-structure of *RapidMiner* is not well-suited for this task. Nevertheless treekernels (and composite kernels) are available in the IE-plugin.
A treekernel proposed by [Collins and Duffy, 2001] is an efficient way to compare

[3] `http://projects.ldc.upenn.edu/ace/`

two trees:

$$K(T_1, T_2) = \sum_i \Big(\sum_{n_1 \in N_1} I_{subtree_i}(n_1) \Big) \Big(\sum_{n_2 \in N_2} I_{subtree_i}(n_2) \Big)$$
$$= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \Delta(n_1, n_2)$$

where T_1 and T_2 are trees and $I_{subtree_i}(n)$ is an indicator-function that returns 1 if the root of subtree i is at node n. $\Delta(n_1, n_2)$ represents the number of common subtrees at node n_1 and n_2. The number of common subtrees finally represents a syntactic similarity measure and can be calculated recursively starting at the leaf nodes in $O(|N_1||N_2|)$.
The composite kernel combines a treekernel and a linear kernel.
[Zhang et al., 2006] therefore present linear combination for instance:

$$K_1(R_1, R_2) = \alpha \hat{K}_L(R_1, R_2) + (1 - \alpha)\hat{K}(T_1, T_2)$$

where R_1 and R_2 are relation-candidates. Tuning the α-value changes the influence of the specific kernels.

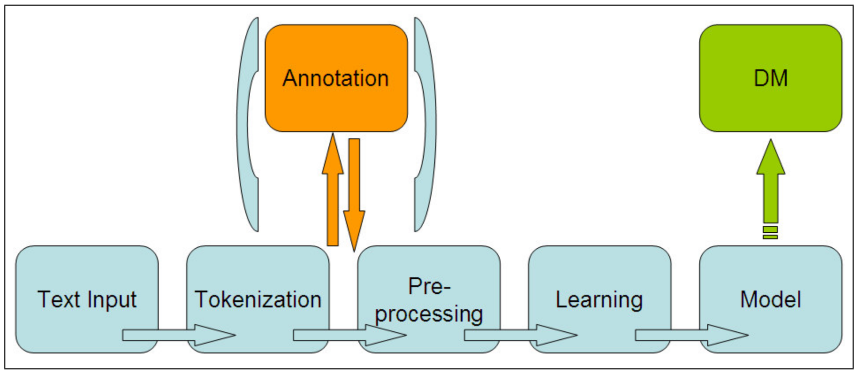


Fig. 2. Schema of the IE-plugin processing

5.3 Evaluation measures

For the evaluation of results for IE-tasks, precision, recall and f-measure are widely used. They are defined as follows:

$$precision = \frac{|\text{correctly labeled tokens}|}{|tokens|}$$
$$recall = \frac{|\text{correctly labeled tokens}|}{|\text{labeled tokens}|}$$
$$f - measure_n = (1 + n)\frac{precision * recall}{n * precision + recall}$$

A special label is reserved for tokens which are 'not labeled'. Labeled tokens have another label but not this special one.

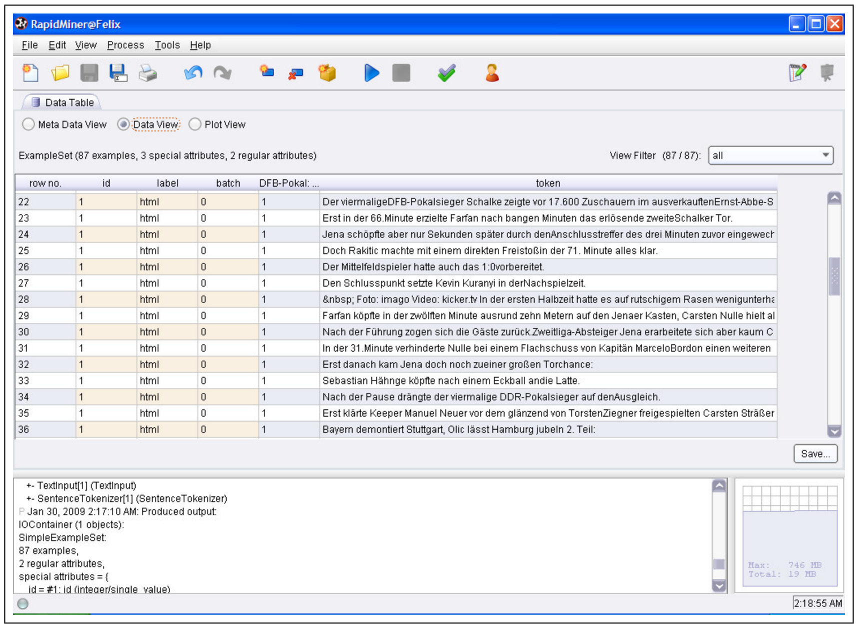
6 Plugin

The IE-task in the plugin is following a simple scheme (Figure 2). The major steps are described in detail in the following part.

Text input and tokenization Texts or documents are available in various forms like plain ascii-files, xml- or html-files and pdf-files for example. Some text-formats are easy to process and some need additional parsing effort. The plugin offers an input-operator which can handle ascii-, html- and pdf-files.
After loading a document, it is present as as a continuous block of text in one table cell, which is not accessible in a profitable way. As a first step tokenizers have to be used to split up the text. Up to now, a trivial sentence- and word-tokenizer is available. Figure 3 shows a html-file in *RapidMiner* after being loaded and tokenized in sentences. While tokenizing splits the texts into smaller parts the original structure is still kept available. So, if one wants to process documents on the word-level, the sentence-level (the level above) is still present (see Figure 5) and can be used for processing the features. These features basically can be extracted from the current position of the document (sentence, word, ...) and from circumfluent positions (see 6).

Annotation and visualization A visualization operator allows to view the documents and to annotate parts of them. One can select attributes which shall be visualized, in order to highlight different aspects of the text. Figure 4 shows a document with some annotated words.

Preprocessing The most important part of the IE-plugin is the preprocessing. The preprocessing-operators are used to enrich the document and its tokens with syntactical information which will later be used to extract semantic information. Tokens can be enriched with contextual information as well as they can deliver inner-token information. One should keep in mind that different tasks need different preprocessing. To use machine learning algorithms for IE, one has to enrich the documents, sentences or words with attributes extracted from themselves (internal evidence) and from their context (external evidence). The plugin offers preprocessing-operators which can be used in a very modular way. There are operators working on document-, sentence- and word-level.
Figure 5 shows a document after tokenization and preprocessing. Every word of the document is represented by one row (example) in the exampleset, so the whole document is contained in the exampleset.
The columns represent the attributes for each example (word). Here, in addition to the word itself (word_0), the prefix of the current word of length 3 (prefix_0_3) is used as an attribute.



Meta Data View [✓ Data View] Plot View

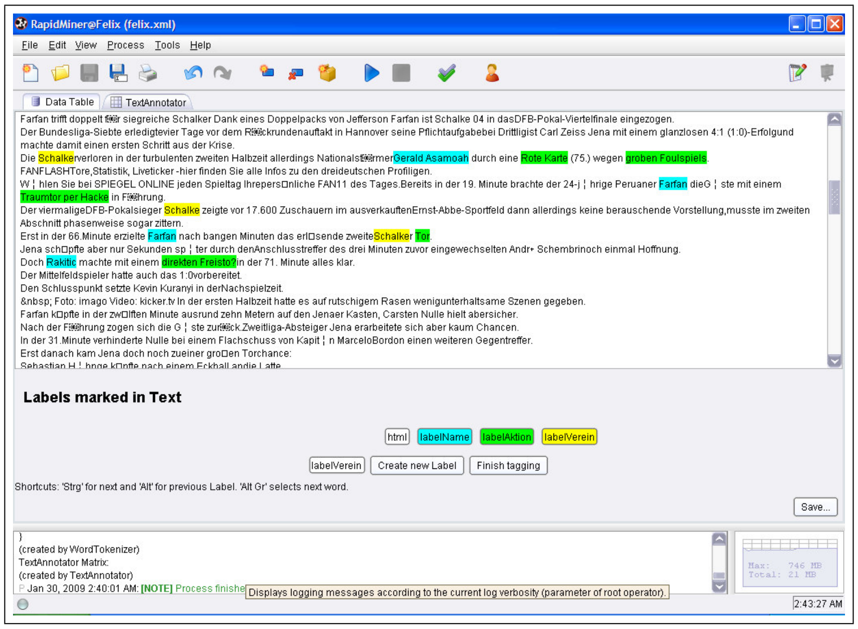
ExampleSet (87 examples, 3 special attributes, 2 regular attributes)

View Filter (87 / 87): all

row no.	id	label	batch	DFB-Pokal: ...	token
22	1	html	0	1	Der viermaligeDFB-Pokalsieger Schalke zeigte vor 17.600 Zuschauern im ausverkauftenErnst-Abbe-S
23	1	html	0	1	Erst in der 66.Minute erzielte Farfan nach bangen Minuten das erlösende zweiteSchalker Tor.
24	1	html	0	1	Jena schöpfte aber nur Sekunden später durch denAnschlusstreffer des drei Minuten zuvor eingewech
25	1	html	0	1	Doch Rakitic machte mit einem direkten Freistoßin der 71. Minute alles klar.
26	1	html	0	1	Der Mittelfeldspieler hatte auch das 1:0vorbereitet.
27	1	html	0	1	Den Schlusspunkt setzte Kevin Kuranyi in derNachspielzeit.
28	1	html	0	1	Foto: imago Video: kicker.tv In der ersten Halbzeit hatte es auf rutschigem Rasen wenigunterha
29	1	html	0	1	Farfan köpfte in der zwölften Minute ausrund zehn Metern auf den Jenaer Kasten, Carsten Nulle hielt al
30	1	html	0	1	Nach der Führung zogen sich die Gäste zurück.Zweitliga-Absteiger Jena erarbeitete sich aber kaum C
31	1	html	0	1	In der 31.Minute verhinderte Nulle bei einem Flachschuss von Kapitän MarceloBordon einen weiteren
32	1	html	0	1	Erst danach kam Jena doch noch zueiner großen Torchance:
33	1	html	0	1	Sebastian Hähnge köpfte nach einem Eckball andie Latte.
34	1	html	0	1	Nach der Pause drängte der viermalige DDR-Pokalsieger auf denAusgleich.
35	1	html	0	1	Erst klärte Keeper Manuel Neuer vor dem glänzend von TorstenZiegner freigespielten Carsten Sträßer
36	1	html	0	1	Bayern demontiert Stuttgart, Olic lässt Hamburg jubeln 2. Teil:

```
  +- TextInput[1] (TextInput)
  +- SentenceTokenizer[1] (SentenceTokenizer)
P Jan 30, 2009 2:17:10 AM: Produced output:
IOContainer (1 objects):
SimpleExampleSet:
87 examples,
2 regular attributes,
special attributes = {
  id = #1: id (integer/single_value)
```

Fig. 3. The html-file after being loaded and tokenized in *RapidMiner*



Farfan trifft doppelt f□r siegreiche Schalker Dank eines Doppelpacks von Jefferson Farfan ist Schalke 04 in dasDFB-Pokal-Viertelfinale eingezogen.
Der Bundesliga-Siebte erledigtevier Tage vor dem R□ckrundenauftakt in Hannover seine Pflichtaufgabebei Drittligist Carl Zeiss Jena mit einem glanzlosen 4:1 (1:0)-Erfolgund machte damit einen ersten Schritt aus der Krise.
Die Schalkeverloren in der turbulenten zweiten Halbzeit allerdings Nationalst□rmerGerald Asamoah durch eine Rote Karte (75.) wegen groben Foulspiels.
FANFLASHTore,Statistik, Liveticker -hier finden Sie alle Infos zu den dreideutschen Profiligen.
W ¦ hlen Sie bei SPIEGEL ONLINE jeden Spieltag Ihrepers□nliche FAN11 des Tages.Bereits in der 19. Minute brachte der 24-j ¦ hrige Peruaner Farfan dieG ¦ ste mit einem Traumtor per Hacke in F□hrung.
Der viermaligeDFB-Pokalsieger Schalke zeigte vor 17.600 Zuschauern im ausverkauftenErnst-Abbe-Sportfeld dann allerdings keine berauschende Vorstellung,musste im zweiten Abschnitt phasenweise sogar zittern.
Erst in der 66.Minute erzielte Farfan nach bangen Minuten das erl□sende zweiteSchalker Tor.
Jena sch□pfte aber nur Sekunden sp ¦ ter durch denAnschlusstreffer des drei Minuten zuvor eingewechselten Andr+ Schembrinoch einmal Hoffnung.
Doch Rakitic machte mit einem direkten Freisto?in der 71. Minute alles klar.
Der Mittelfeldspieler hatte auch das 1:0vorbereitet.
Den Schlusspunkt setzte Kevin Kuranyi in derNachspielzeit.
  Foto: imago Video: kicker.tv In der ersten Halbzeit hatte es auf rutschigem Rasen wenigunterhaltsame Szenen gegeben.
Farfan k□pfte in der zw□lften Minute ausrund zehn Metern auf den Jenaer Kasten, Carsten Nulle hielt abersicher.
Nach der F□hrung zogen sich die G ¦ ste zur□ck.Zweitliga-Absteiger Jena erarbeitete sich aber kaum Chancen.
In der 31.Minute verhinderte Nulle bei einem Flachschuss von Kapit ¦ n MarceloBordon einen weiteren Gegentreffer.
Erst danach kam Jena doch noch zueiner gro□en Torchance:
Sebastian H ¦ hnge k□pfte nach einem Eckball andie Latte.

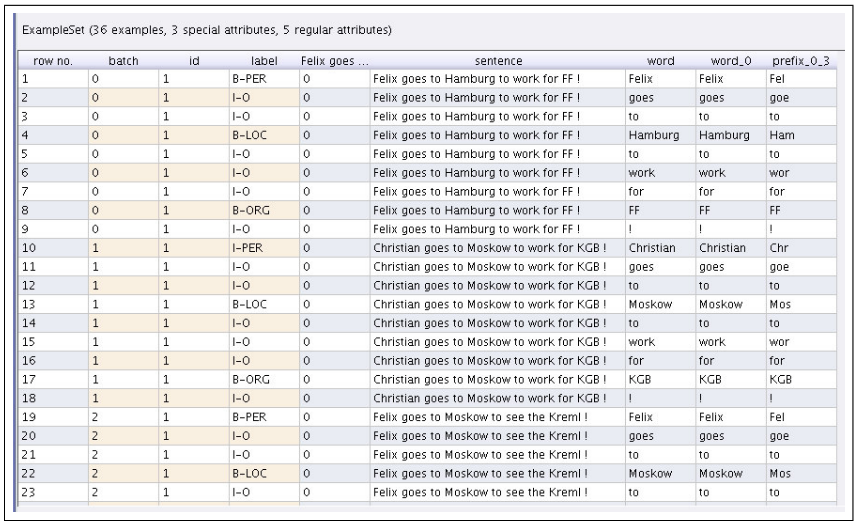
Labels marked in Text

Shortcuts: 'Strg' for next and 'Alt' for previous Label. 'Alt Gr' selects next word.

```
}
(created by WordTokenizer)
TextAnnotator Matrix:
(created by TextAnnotator)
P Jan 30, 2009 2:40:01 AM: [NOTE] Process finishe
```

Displays logging messages according to the current log verbosity (parameter of root operator).

Fig. 4. The html-file after being tokenized in words and annotated



ExampleSet (36 examples, 3 special attributes, 5 regular attributes)

row no.	batch	id	label	Felix goes ...	sentence	word	word_0	prefix_0_3
1	0	1	B-PER	0	Felix goes to Hamburg to work for FF !	Felix	Felix	Fel
2	0	1	I-O	0	Felix goes to Hamburg to work for FF !	goes	goes	goe
3	0	1	I-O	0	Felix goes to Hamburg to work for FF !	to	to	to
4	0	1	B-LOC	0	Felix goes to Hamburg to work for FF !	Hamburg	Hamburg	Ham
5	0	1	I-O	0	Felix goes to Hamburg to work for FF !	to	to	to
6	0	1	I-O	0	Felix goes to Hamburg to work for FF !	work	work	wor
7	0	1	I-O	0	Felix goes to Hamburg to work for FF !	for	for	for
8	0	1	B-ORG	0	Felix goes to Hamburg to work for FF !	FF	FF	FF
9	0	1	I-O	0	Felix goes to Hamburg to work for FF !	!	!	!
10	1	1	I-PER	0	Christian goes to Moskow to work for KGB !	Christian	Christian	Chr
11	1	1	I-O	0	Christian goes to Moskow to work for KGB !	goes	goes	goe
12	1	1	I-O	0	Christian goes to Moskow to work for KGB !	to	to	to
13	1	1	B-LOC	0	Christian goes to Moskow to work for KGB !	Moskow	Moskow	Mos
14	1	1	I-O	0	Christian goes to Moskow to work for KGB !	to	to	to
15	1	1	I-O	0	Christian goes to Moskow to work for KGB !	work	work	wor
16	1	1	I-O	0	Christian goes to Moskow to work for KGB !	for	for	for
17	1	1	B-ORG	0	Christian goes to Moskow to work for KGB !	KGB	KGB	KGB
18	1	1	I-O	0	Christian goes to Moskow to work for KGB !	!	!	!
19	2	1	B-PER	0	Felix goes to Moskow to see the Kreml !	Felix	Felix	Fel
20	2	1	I-O	0	Felix goes to Moskow to see the Kreml !	goes	goes	goe
21	2	1	I-O	0	Felix goes to Moskow to see the Kreml !	to	to	to
22	2	1	B-LOC	0	Felix goes to Moskow to see the Kreml !	Moskow	Moskow	Mos
23	2	1	I-O	0	Felix goes to Moskow to see the Kreml !	to	to	to

Fig. 5. Exampleset after preprocessing

Learning The IE-plugin offers operators for NER and for RE. *RapidMiner* learners deliver so called models which equate to a function and can be applied to an (unlabeled) exampleset. Until now, for NER, the Conditional Random Fields (CRF) operator (see section 5.1) can be used. For RE the treekernel operator (see section 5.2) should be used. The underlying techniques have been described before. The implementation of these learning algorithms has been kept modular to allow the combination of various methods. Due to this modularization the CRF-learner can be combined with various optimization-methods such as quasi-newton-methods or evolutionary algorithms(particle swarm optimization for example).

DM-usage The calculated model can – of course – be used for the extraction of interesting information from unseen documents, but after having processed entities or relations or every other information one can easily build up a new exampleset containing the extracted information to gain additional knowledge.

7 Conclusion and Future Work

We presented the IE-plugin[4] for the opensource datamining software *Rapid-Miner*. The IE-plugin enriches the functionalities of *RapidMiner* with IE-related techniques.
The flat internal database-like data-format has the advantage of easily allowing flat and simple syntactical features to be added or removed. A very broad and

[4] the plugin should be available at `http://rapid-i.com/content/view/55/85/`

deep access to the data and the learners is possible. Because of the modular architecture it is possible to change many settings – in contrast to black-boxed systems.
A minor disadvantage of the current implementation is, that structured information can not be handled in an easy and adequate way, yet.Therefore the next steps will be to develop new internal datatypes to handle structural information in a better way. Additionally, to evaluate the performance of the implemented IE operators currently tests are running on well known datasets for RE to compare the results of the plugin with other systems.

References

[LDC, 2004] (2004). *The ACE 2004 Evaluation Plan.* Linguistic Data Consortium.

[Cardie, 1997] Cardie, C. (1997). Empirical methods in information extraction. *AI Magazine*, 18.

[Chapman et al., 1999] Chapman, P., Clinton, J., Khabaza, T., Reinartz, T., and Wirth, R. (1999). The crisp–dm process model. Technical report, The CRIP–DM Consortium NCR Systems Engineering Copenhagen, DaimlerChrysler AG, Integral Solutions Ltd., and OHRA Verzekeringen en Bank Groep B.V. This Project (24959) is partially funded by the European Commission under the ESPRIT Program.

[Collins and Duffy, 2001] Collins, M. and Duffy, N. (2001). Convolution kernels for natural language. In *Proceedings of Neural Information Processing Systems, NIPS 2001.*

[Collins and Singer, 1999] Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.*

[Etzioni et al., 2005] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. In *Artificial Intelligence*, volume 165, pages 91 – 134. Elsevier Science Publishers Ltd. Essex, UK.

[Grishman, 2003] Grishman, R. (2003). Information extraction. In *Handbook of Computational Linguistics Information Extraction*, chapter 30. Oxford University Press, USA.

[Grishman and Sundheim, 1996] Grishman, R. and Sundheim, B. (1996). Message understanding conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING).*

[Haussler, 1999] Haussler, D. (1999). Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz, Department of Computer Science, Santa Cruz, CA 95064, USA.

[Kim et al., 2004] Kim, J.-D., Otha, T., Yoshimasa, T., Yuka, T., and Collier, N. (2004). Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications(JNLPBA-2004)*, Geneva, Switzerland.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

[Liu and Nocedal, 1989] Liu, D. C. and Nocedal, J. (1989). On the limited memory method for large scale optimization. In *Mathematical Programming*, volume 45, pages 503–528. Springer Berlin / Heidelberg.

[McCallum et al., 2000] McCallum, A., Freitag, D., and Pereira, F. (2000). Maximum Entropy Markov Models for information extraction and segmentation. In *Proc. 17th International Conf. on Machine Learning*, pages 591–598. Morgan Kaufmann, San Francisco, CA.

[McDonald, 1996] McDonald, D. (1996). Internal and external evidence in the identification and semantic categorization of proper names. In Boguraev, B. and Pustejovsky, J., editors, *Corpus Processing for Lexical Acquisition*, pages 21–39. MIT Press, Cambridge, MA.

[Mierswa et al., 2006] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., and Euler, T. (2006). YALE: Rapid Prototyping for Complex Data Mining Tasks. In Eliassi-Rad, T., Ungar, L. H., Craven, M., and Gunopulos, D., editors, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pages 935–940, New York, USA. ACM Press.

[Nadeau et al., 2006] Nadeau, D., Turney, P. D., and Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286.

[Tsochantaridis et al., 2005] Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research*, 6:1453–1484.

[Zhang et al., 2007] Zhang, M., Che, W., Aw, A. T., Tan, C. L., Zhou, G., Liu, T., and Li, S. (2007). A grammar-driven convolution tree kernel for semantic role classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 200–207. Association for Computational Linguistics.

[Zhang et al., 2006] Zhang, M., Zhang, J., Su, J., and Zhou, G. (2006). A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings 44th Annual Meeting of ACL*, pages 825–832.

Abbreviations

CRF	Conditional Random Fields
DM	Data Mining
IE	Information Extraction
KDD	Knowledge Discovery in Databases
ML	Machine Learning
MUC	Message Understanding Conference
NE	Named Entity
NER	Named Entity Recognition
RE	Relation Extraction