

Ontology-driven Event Recognition on Stories

Maria Vargas-Vera

Knowledge Media Institute (KMi),
The Open University,
Walton Hall, Milton Keynes, MK7 6AA,
United Kingdom
m.vargas-vera@open.ac.uk

David Celjuska

Department of Artificial Intelligence
and Cybernetics
Technical University of Kosice,
Letna 9/A, 04001 Kosice, Slovakia
celjuska@neuron.tuke.sk

Abstract

This paper describes a system which recognizes events on stories. Our system classifies stories and populates a KMi Planet ontology with new instances of classes defined in it. Currently, the system recognizes events which can be classified as belonging to a single category and it also recognizes overlapping events (more than one event is recognized in the story). In each case, the system provides a confidence value associated to the suggested classification. In our event recognition system we use Information Extraction and Machine Learning technologies. We have tested this system using an archive of stories describing the academic life of our institution.

1 Introduction

In this paper we focus on the problem of automatically classifying documents. This is an interesting problem in Natural Language research and it has many potential applications ranging from document summarization to the semantic web. There are several approaches to text classification. In this paper we describe an approach to stories classification based on Information Extraction and Machine Learning technologies. Essentially, information extraction can be seen as the task of pulling predefined relations from texts. Efforts have been made to apply information extraction to several domains, for instance, scientific articles such as MEDLINE (Craven and Kuhlman, 1999), bibliographic notices (Michell, 1997) and medical records (Soderland et al., 1995). In designing an information extraction system for the KMi organization, the system should be able to extract the name of KMi projects, KMi funding organizations, awards, dates, etc. and ignore anything not clearly relevant to these pre-specified categories. Ontologies can be used in information extraction systems to help them extract

relations from semi- or unstructured documents, statements or terms (Roux et al., 2000). Also, recent work on semi-automatic ontology acquisition by means of information extraction, supported by machine-learning methods, is described in (Maedche and Staab, 2000; Kietz et al., 2000; Vargas-Vera et al., 2001a; Vargas-Vera et al., 2001b). On similar lines, there is CMU's approach for extracting information from hypertext using machine learning techniques and making use of an ontology (Craven et al., 1999).

Our system, as most information extraction systems, uses some form of partial parsing to recognize syntactic constructs without generating a complete parse tree for each sentence. Such partial parsing has the advantages of greater speed and robustness. High speed is necessary to apply the information extraction to a large set of documents. The robustness achieved by allowing useful work to be done from a partial parsing is essential to deal with unstructured and informal texts.

The main contribution of our paper can be summarized as follows:

- identification of events on stories by means of Information Extraction and Machine Learning technology and
- semi-automatic population of a selected ontology.

The paper is organized as follows: Section 2 shows the event topology used in our event recognition system. Section 3 describes the classification of stories. Section 4 presents the process model in our system. Section 5 presents assignment of confidence values to the rules extracted using Crystal. Section 6 describes semi-automatic population of a selected ontology with new instances of classes already defined in our selected ontology. Section 7 presents a working example. Section 8 presents the evaluation carried out using the KMi archive of stories. Finally, Section 9 gives conclusions and directions for future work.

2 Event topology

The KMi domain consists of events or activities happening in our Institute. Events are defined formally in our ontology as classes. Currently, in the KMi ontology we have defined 41 different types of events. As the event topology is already defined in the ontology. Then, for each event we already have defined the slots which might be instantiated by an information extraction component. Figure 1 shows a portion of the hierarchy of events as defined in the ontology.

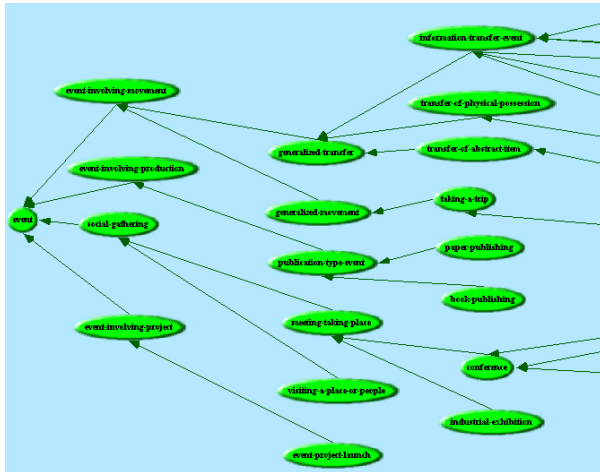


Figure 1: Event hierarchy

For the sake of space, we only present the structure of three type of events from the event hierarchy: visiting-a-place-or-people, project-award and academic-conference.

Class Event 1: visiting-a-place-or-people

```
slots:
  visitor (list of person(s))
  people-or-organisation-being-visited
    (list of person(s) or organization)
  has-duration (duration)
  start-time (time-point)
  end-time (time-point)
  has-location (a place)
  other agents-involved
    (list of person(s))
  main-agent (list of person(s))
```

The structure of Event 1 (visiting-a-place-or-people) describes a set of objects which might be encountered in story describing an event visit, such as visitor, people-or-organisation-being-visited, other agents-involved, etc.

Class Event 2: project-award

```
slots:
  has-awarding-body (an organization)
  has-award-rationale (project goals)
  object-acted-on (award)
  recipient-agents (the agents which
    possibly receive
    the object-acted-on)
  start-time (time-point)
  end-time (time-point)
  location-at-start (a place)
  location-at-end (a place)
```

The structure of Event 2 describes a set of objects which might be encountered in story describing an event “project-award” such as organization, award, recipient agent, etc.

Class event 3: academic-conference

```
slots:
  has-papers (list of papers)
  has-invited-talks (list of talk(s))
  has-demonstrations
    (list of demonstration(s))
  has-duration (duration)
  start-time (time-point)
  end-time (time-point)
  has-location (a place)
  main-agent (list of person(s))
  other agents-involved
    (list of person(s))
  meeting-attendees (list of person(s))
  meeting-organizer (list of person(s))
```

Event 3 contains the structure for the event “academic-conference”. Entities that need to be recognized are papers, location, etc.

3 Classification of stories

We classify stories or documents as belonging to any of the types of events according with the objects that are found in them. For each event type we have a predefined objects that should be found in the story. For instance, for the event “visiting-a-place-or-people” the system might encounter objects of type: visitor, place and date.

In our system classification is performed in the following steps:

- pre-process the story
- find the objects in the story using partial parsing
- provide classification of a story with associated confidence value

Each event in our system has several patterns which can be used to recognize it. For instance, in case of

“visiting-place-or-people” event the following patterns were encountered.

- X visited Y
- X visits Y
- Y visited by X
- Y visited by X at Z
- Y were visited by X
- Visit to Y
- X came
- X came to visit Y
- Y hosted X
- Y hosted a visit from X
- Y had a visit from X
- Y welcomes X

In all patterns shown above X is a person, Y is a place/institution and Z is a location.

Problems might occur when more than one event can be recognized in a story. Then our system decides to classify the story according with the following criteria:

it computes the confidence value as the number of slots the system was able to extract divided by total number of slots that an annotator/expert used during annotation process on any story from a given class.

The confidence of classification into a given class is shown below.

$$\text{confidence_classification} = \frac{\text{number of items extracted}}{\text{number of slots used by annotator}}$$

Then, the category which maximizes the sum of the filled slots is placed at the top of the window (i.e. the classification with the maximum confidence value). If none of the templates are able to be filled (during the extraction phase) then the story is given the status of unclassified story. The user will be presented with classification, associated confidence value and extracted objects. Once that the user agrees (rejects) one (all) of the suggested classification and extracted information, the ontology is updated with a new instance.

4 Process model

Within this work we have focused on creating a generic process model for event recognition on stories. In our system we have devised three activities: mark-up, learning and extraction. We will provide more details of each of the activities in turn.

Mark-up

The activity of semantic tagging refers to the activity of annotating text documents (written in plain ASCII or HTML) with a set of tags defined in the ontology. Our classification system provides means to browse the event hierarchy. In this hierarchy each event is a class and the annotation component extracts the set of possible tags

from the slots defined in the ontology. During the mark-up phase as the text is selected the system inserts the relevant SGML tags into the document. Also our system offers the possibility of removing tags from a document.

Learning

This phase was implemented by integrating two tools: Marmot and a learning component called Crystal, both from UMass (full description can be found in (Riloff, 1996a)). Marmot is a natural language preprocessing tool that accepts ASCII files and produces an intermediate level of text analysis that is useful for information extraction applications. Sentences are separated and segmented into noun phrases, verb phrases prepositional phrases. Marmot has several functionalities: preprocesses abbreviations to guide sentence segmentation, resolves sentences boundaries, identifies parenthetical expressions, recognizes entries from a phrasal lexicon and replace them, recognizes dates and duration phrases, performs phrasal bracketing of noun, preposition and adverbial phrases, finally scopes conjunctions and disjunctions.

Crystal is a dictionary induction tool. It derives a dictionary of concept nodes from a training corpus. The first step in dictionary creation is the annotation of a set of training texts by a domain expert. Each phrase that contains information to be extracted is tagged (with SGML style tags).

Crystal initializes a concept nodes dictionary for each positive instance of each type of event. The initial concept node definitions are designed to extract the relevant phrases in the training instance that creates them but are too specific to apply to a unseen sentences. The main task of Crystal is to gradually relax the constraints on the initial definitions and also to merge similar definitions. Crystal finds generalizations of its initial concept node definitions by comparing definitions that are similar. This similarity is deduced by counting the number of relaxations required to unify two concept node definitions. Then a new definition is created with constraints relaxed. Finally the new definition is tested against the training corpus to insure that it does not extract phrases that were not marked with the original two definitions. This means that Crystal takes similar instances and generalizes into a more general rule by preserving the properties from each of the concept node definitions which are generalized.

The inductive concept learning in Crystal is similar to the inductive learning algorithm described in (Michell, 1982) a specific-to-general data-driven search to find the most specific generalization that covers all positive instances. Crystal finds the most specific generalization that covers all positive instances but uses a greedy unification of similar instances rather than breadth-first search.

Extraction

A third component called Badger (from UMass) was also integrated into our event recognition system. Badger makes the instantiation of templates. The main task of badger is to take each sentence in the text and see if it matches any of our concept node definitions. If no extraction concept node definition applies to a sentence, then no information will be extracted; this means that irrelevant text can be processed very quickly.

It might occur that Badger obtains more than one type of event for an story¹. Then our information extraction system decides to classify the story according with the criteria defined in section 3.

5 Confidence values associated to the extraction rules

In the automatic construction of ontologies precision is more important than recall. Therefore, our goal is to obtain high precision. Currently we are focusing, in associating a confidence value to the Crystal induced rules in order to increase precision. In our solution we have associated confidence value to the Crystal induced rules. The confidence number for each rule can be computed by a three-fold cross-validation methodology on the training set. According to this methodology, the training set is split into three equally sized subsets and the learning algorithm is run three times. Each time two of the three pieces are used for training and the third is kept as unseen data (test set) for the evaluation of the induced rules. The final result is the average over the three runs. At run time each instance extracted by Badger will be assigned the precision value of that rule. The main feature of using confidence values is that among ambiguous instantiations, we can still choose the one with the highest estimated confidence.

In the new scenario we have a set of rules for each event like the ones shown below.

```
- Rule_1           (confidence_1)
- Rule_2           (confidence_2)
- Rule_3           (confidence_3)
- .               .
- .               .
- .               .
- Rule_(n-1)       (confidence_(n-1))
- Rule_n           (confidence_n)
```

6 Populating the ontology

Building domain-specific ontologies often requires time-consuming expensive manual construction. Therefore,

¹The first implementation of our event recognition system which only recognizes single events is described in (Vargas-Vera et al., 2001b)

we envisage information extraction as a technology that might help us during ontology maintenance process. During the population step our information extraction system has to fill predefined slots associated with each event, as already defined the ontology. Our goal is to automatically fill as many slots as possible. However, some of the slots will probably still require manual intervention. There are several reasons for this problem:

- there is information that is not stated in the story,
- none of our patterns match with the sentence that might provide the information (incomplete library of patterns)

The extracted information could be validated using the ontology. This is possible because each slot of each class of the ontology has a type associated with it. Therefore, extracted information which does not match the type definition of the slot in the ontology can be highlighted as incorrect.

7 Example

In this section we present an example. The domain of our example is a web based news letter, KMi Planet (Domingue and Scott, 2000) that has been running in our institution for five years. One of the functionalities offered in KMi Planet is an editor which can be used to manually extract information and classify a story. Then this information is sent to ontology server in order to create a new instance. Whilst we are happy with the functionalities offered by KMi planet, we want to automatically extract and populate the handcrafted ontology. This is because maintenance is time consuming and error prone. Therefore, in the first instance we have selected KMi Planet (which contains 200 stories of academic life in KMi) as our first domain for our event recognition system.

In this section we show two examples in the first story only one single event was recognized and the second example two events were recognized in a story. Figure 2 shows an story classified as “project-award” with a confidence of 75% whilst in Figure 3 our system suggest two classifications for the story “visiting-a-place-or-people” and “project-award” with confidence values of 33% and 25% respectively.

In Figure 2 we can see in the top right window that the user has selected the project-award class from the KMi ontology. Also in the same figure the number 3/4 means that three out of four slots values were able to be extracted from story. The number in brackets is the confidence value associated to the suggested classification. For this particular story titled “The AKT begins” our system was able to extract the object-acted-on (thing that was given as an award), recipient-agent

(man/organization/project that was given an award) and has-awarding-body (man/organization that gave award).

In Figure 3 our system was able to extract visitor for classification “visiting-a-place-or-people” and recipient-agent for classification “project-award”. The reason why this story is classified as project-award is that during the learning phase the system learn a rule:

```
Cn-type project-award          ID:18
      Status: GENERALIZED
```

Constraints:

VB::

```
mode active
root: bring
terms: BRINGS
mod terms: <null>
head terms:BRINGS
```

```
classes: ws_Root_Class
mod class: ws_Root_Class
```

PP:: ==> recipient-agents

```
terms: TO
mod terms: TO
classes: ws_Root_Class
mod class: ws_Root_Class
head class: ws_Root_Class
```

As we can see this rule is very general and the system from any sentence with verb=BRING and preposition=TO extracts Prepositional phrase(PP) as recipient agent. This problem could be solved by constraining slots to the correct type given in an ontology.

Finally, after information is extracted the classification and extracted values are presented to the user. If the user accept these values then the ontology will be updated with a new instance.

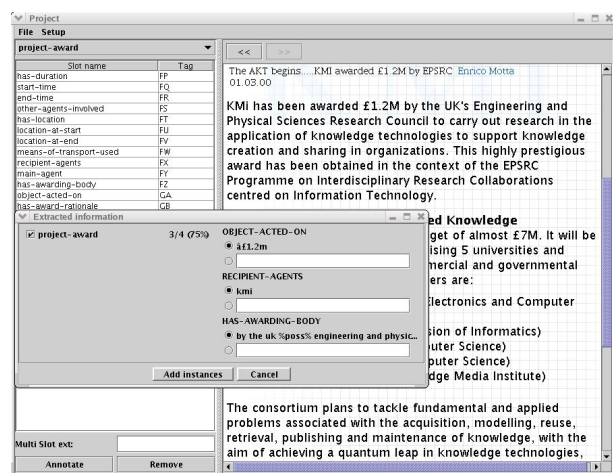


Figure 2: One suggested classification for story titled “The AKT begins”

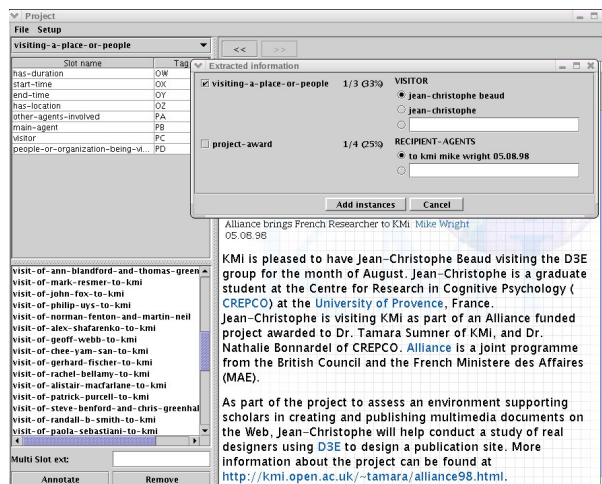


Figure 3: Two suggested classifications for a story titled “Alliance brings French researcher to KMi”

8 Evaluation

We have tested our event recognition system performing three-fold cross-validation methodology. In this evaluation we have used standard metrics for computing precision and recall²

Previous work has reported that spurious patterns were deleted manually from the library of rules under the assumption that they were not likely to be of much value (Riloff, 1996b). However, our experiments were carried out without deleting spurious rules.

Events	Trial 1		Trial 2		Trial 3	
	Prec	Rec	Prec	Rec	Prec	Rec
Visiting-a-pl	0.77	0.47	0.82	0.47	0.47	0.63
project-awa	0.89	0.35	0.50	0.15	0.46	0.36

Table 1: Precision and Recall for 2 events

The trials for “visiting-a-place-or-people” and “project-award” events are presented in Table 1.

Overall precision for event “visiting-a-place-or-people” was 0.68 and overall recall is 0.52. The experimental results suggested that precision could drop dramatically if the set of extraction rules is used as generated. We expect that by associating confidence value to the extraction

² As a reminder, precision and recall metrics are defined as follows:

$$Precision = \frac{\text{number of items correctly extracted}}{\text{total number of items extracted by system}}$$

$$Recall = \frac{\text{number of items correctly extracted}}{\text{total number of items need to be extracted by system}}$$

rules the performance of the event recognition system will be increased. Therefore, further research needs to be undertaken in the direction of the association of a confidence to the extraction rules.

We have also experimented with MnM (Vargas-Vera et al., 2002) using as testbed the KMi stories. MnM has integrated as information extraction Amilcare. In a single-slot Engine such as Amilcare each rule is applied independently of others. This causes overlapping matches among different rules resulting in multiple ambiguous instantiations for those matches. Therefore, we have to resolve those ambiguities and choose the correct instantiations. This ambiguity might be removed by performing post-processing capable of discarding wrong instantiations of slots. However, by using confidence value associated to the extraction rules we will try to overcome this problem.

9 Conclusions and future work

We have built a tool which recognizes events on stories and extracts knowledge using an ontology. Currently, our system have been trained using the archive of 200 stories that we have collected in KMi.³ The training step was performed using typical examples of stories belonging to each of the different type events defined in the ontology. Our system recognizes single events and overlapping events. Then it is able to suggest possible classification for a story. Currently, the population of the selected ontology is performed at the level of instances. Our system extracts instances of classes defined in the event ontology. However, in future we will explore the possibility of use the extracted information with conceptool (Compatangelo and Meisel, 2003) in order to create new classes in a selected ontology. This will allow us to refine our ontology with a finer granularity.

The experiments showed that an automatic mechanism is needed in order to determine which extraction rules are spurious. We believe this problem can be solved by associating confidence value to the extraction rules.

As a medium term goal, we plan to link our event recognition system with KMi research profiles. Then we will filter stories and send them by e-mail to the people who might be interested in reading them. In this way, KMi researchers will be informed about events in our institution without having to browse the archive of news.

Currently, our event recognition system works with the KMi Planet ontology. But, in future, we plan to offer a selection of ontologies.

Acknowledgments

This work was funded by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration

³URL:<http://kmi.open.ac.uk/planet/>

(IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N157764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University. The authors would like to thank Mark Gaved for his invaluable help in reviewing the first draft of this paper.

References

- E. Compatangelo and H. Meisel. 2003. Reasonable support to knowledge sharing through schema analysis and articulation. *Journal of Engineering Intelligent Systems*.
- M. Craven and J. Kumlien. 1999. Constructing Biological Knowledge Bases by Extracting Information from Text Sources. Proceedings of The 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99).
- M. Craven and D. DiPasquo and D. Freitag and A. McCallum and T. Mitchell, K. Nigam and S. Slattery. 1999. Learning to Construct Knowledge Bases from the World Wide Web. *Journal Artificial Intelligence*.
- J. Domingue and P. Scott. 2000. KMi Planet: A Web Based News Server. In proceedings of the Asia Pacific Computer Human Interaction Conference (APCHI-98).
- Kietz, J.-U. and Maedche, A. and Volz, R. 2000. A method for Semi-automatic ontology acquisition from a corporate intranet. Proceedings of the EKAW'00 Workshop on Ontologies and Text, Juan-Les-Pins, France.
- Maedche A. and S. Staab. 2000. Semi-automatic engineering of ontologies from texts. Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, SEKE2000, Chicago, IL, USA pages 231-239.
- T. Mitchell. 1982. Generalization as search. *Journal Artificial Intelligence*, volume 18:203-226.
- D. Proux and Y. Chenevoy 1997. Natural Language Processing for Book Storage: Automatic Extraction of Information from Bibliographic Notices Proceedings of The Natural Language Processing Pacific Rim Symposium (NLPRS'97), Thailand, December 2-4, pages 229-234.
- E. Riloff. 1996a. An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *AI Journal*, volume 85:101-134.
- E. Riloff. 1996b. Automatically Generating Extracting Patterns from Untagged Text. Proceedings of the Thirtieth National Conference on Artificial Intelligence (AAAI-96), pp 1044-1049.

- C. Roux and D. Proux and F. Rechenmann and L. Julliard. 2000. An Ontology Enrichment Method for a Pragmatic Information Extraction System gathering Data on Genetic Interactions. Proceedings of The 14th European Conference on Artificial Intelligence (Workshop on Ontology Learning ECAI-2000), Berlin, Germany, August 20, 2000.
- S. Soderland and D. Aronow and D. Fisher and J. Aseltine and W. Lehnert. 1995. Machine Learning of Text Analysis Rules for Clinical Records Center for Intelligent Information Retrieval, TR 39.
- M. Vargas-Vera and J. Domingue and Y. Kalfoglou and E. Motta and S. Buckingham Shum. 2001a. Template-Driven Information Extraction for Populating Ontologies. In proceedings of the Workshop Ontology Learning IJCAI-2001, Seattle, USA.
- M. Vargas-Vera and J. Domingue and E. Motta and S. Buckingham Shum and M. Lanzoni. 2001b. Knowledge Extraction by using an Ontology-based Annotation Tool. In proceedings of the Workshop Knowledge Markup & Semantic Annotation, held in conjunction with the First International Conference on Knowledge Capture (K-CAP 2001), Victoria Canada, pages 5-12.
- M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt and F. Ciravegna. 2002. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. The 13th International Conference on Knowledge Engineering and Management (EKAW 2002), Lecture Notes in Computer Science 2473, ed Gomez-Perez, A., Springer Verlag, 2002, 379-391, ISBN 3-540-44268-5.