Unsupervised Relation Extraction from Web Documents

Kathrin Eichler, Holmer Hemsen and Günter Neumann

DFKI GmbH, LT-Lab, Stuhlsatzenhausweg 3 (Building D3 2), D-66123 Saarbrücken
{FirstName.SecondName}@dfki.de

Abstract

The IDEX system is a prototype of an interactive dynamic Information Extraction (IE) system. A user of the system expresses an information request in the form of a topic description, which is used for an initial search in order to retrieve a relevant set of documents. On basis of this set of documents, unsupervised relation extraction and clustering is done by the system. The results of these operations can then be interactively inspected by the user. In this paper we describe the relation extraction and clustering components of the IDEX system. Preliminary evaluation results of these components are presented and an overview is given of possible enhancements to improve the relation extraction and clustering components.

1. Introduction

Information extraction (IE) involves the process of automatically identifying instances of certain relations of interest, e.g., produce(<company>, <product>, <location>), in some document collection and the construction of a database with information about each individual instance (e.g., the participants of a meeting, the date and time of the meeting). Currently, IE systems are usually domain-dependent and adapting the system to a new domain requires a high amount of manual labour, such as specifying and implementing relation–specific extraction patterns manually (cf. Fig. 1) or annotating large amounts of training corpora (cf. Fig. 2). These adaptations have to be made offline, i.e., before the specific IE system is actually made. Consequently, current IE technology is highly statical and inflexible with respect to a timely adaptation to new requirements in the form of new topics.

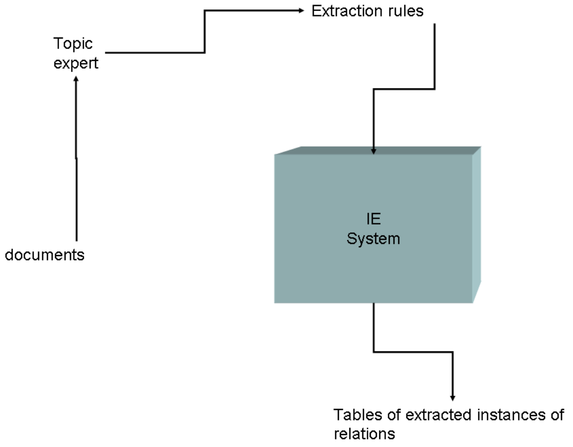

Figure 1: A hand-coded rule–based IE–system (schematically): A topic expert implements manually task–specific extraction rules on the basis of her manual analysis of a representative corpus.

1.1. Our goal

The goal of our IE research is the conception and implementation of core IE technology to produce a new IE system automatically for a given topic. Here, the pre–knowledge about the information request is given by a user online to the IE core system (called IDEX) in the form of a topic description (cf. Fig. 3). This initial information source is used to retrieve relevant documents and extract and cluster relations in an unsupervised way. In this way, IDEX is able to adapt much better to the dynamic information space, in particular because no predefined patterns of relevant relations have to be specified, but relevant patterns are determined online. Our system consists of a front-end, which provides the user with a GUI for interactively inspecting information extracted from topic-related web documents, and a back-end, which contains the relation extraction and clustering component. In this paper, we describe the back-end component and present preliminary evaluation results.

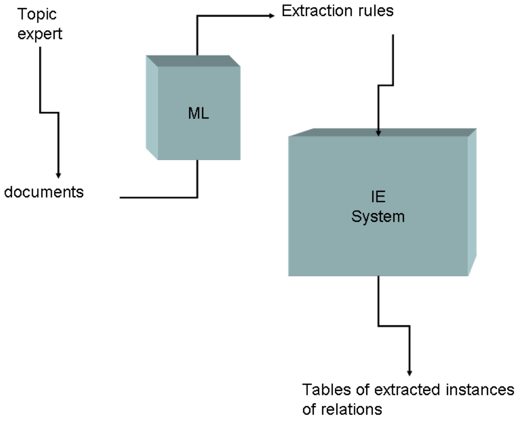

Figure 2: A data–oriented IE system (schematically): The task–specific extraction rules are automatically acquired by means of Machine Learning algorithms, which are using a sufficiently large enough corpus of topic–relevant documents. These documents have to be collected and costly annotated by a topic–expert.

1.2. Application potential

However, before doing so we would like to motivate the application potential and impact of the IDEX ap-

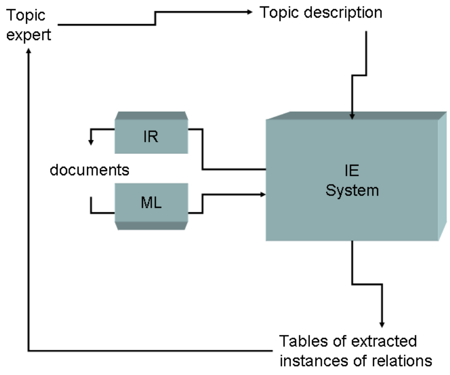

Figure 3: The dynamic IE system IDEX (schematically): a user of the IDEX IE system expresses her information request in the form of a topic description which is used for an initial search in order to retrieve a relevant set of documents. From this set of documents, the system extracts and collects (using the IE core components of IDEX) a set of tables of instances of possibly relevant relations. These tables are presented to the user (who is assumed to be the topic–expert), who will analyse the data further for her information research. The whole IE process is dynamic, since no offline data is required, and the IE process is interactive, since the topic expert is able to specify new topic descriptions, which express her new attention triggered by a novel relationship she was not aware of beforehand.

proach by an example application. Consider, e.g., the case of the exploration and the exposure of corruptions or the risk analysis of mega construction projects. Via the Internet, a large pool of information resources of such mega construction projects is available. These information resources are rich in quantity, but also in quality, e.g., business reports, company profiles, blogs, reports by tourists, who visited these construction projects, but also web documents, which only mention the project name and nothing else. One of the challenges for the risk analysis of mega construction projects is the efficient exploration of the possibly relevant search space. Developing manually an IE system is often not possible because of the timely need of the information, and, more importantly, is probably not useful, because the needed (hidden) information is actually not known. In contrast, an unsupervised and dynamic IE system like IDEX can be used to support the expert in the exploration of the search space through pro–active identification and clustering of structured entities. Named entities like for example person names and locations, are often useful indicators of relevant text passages, in particular, if the names are in some relationship. Furthermore, because the found relationships are visualized using an advanced graphical user interface, the user can select specific names and find associated relationships to other names, the documents they occur in or she can search for paraphrases of sentences.

2. System architecture

The back-end component, visualized in Figure 4, consists of three parts, which are described in detail in this section: preprocessing, relation extraction and relation clustering.

2.1. Preprocessing

In the first step, for a specific search task, a topic of interest has to be defined in the form of a query. For this topic, documents are automatically retrieved from the web using the Google search engine. HTML and PDF documents are converted into plain text files. As the tools used for linguistic processing (NE recognition, parsing, etc.) are language-specific, we use the Google language filter option when downloading the documents. However, this does not prevent some documents written in a language other than our target language (English) from entering our corpus. In addition, some web sites contain text written in several languages. In order to restrict the processing to sentences written in English, we apply a language guesser tool, lc4j (Lc4j, 2007) and remove sentences not classified as written in English. This reduces errors on the following levels of processing. We also remove sentences that only contain non-alphanumeric characters. To all remaining sentences, we apply LingPipe (LingPipe, 2007) for sentence boundary detection, named entity recognition (NER) and coreference resolution. As a result of this step database tables are created, containing references to the original document, sentences and detected named entities (NEs).

2.2. Relation extraction

Relation extraction is done on the basis of parsing potentially relevant sentences. We define a sentence to be of potential relevance if it at least contains two NEs.
In the first step, so-called skeletons (simplified dependency trees) are extracted. To build the skeletons, the Stanford parser (Stanford Parser, 2007) is used to generate dependency trees for the potentially relevant sentences. For each NE pair in a sentence, the common root element in the corresponding tree is identified and the elements from each of the NEs to the root are collected. An example of a skeleton is shown in Figure 5.
In the second step, information based on dependency types is extracted for the potentially relevant sentences. Focusing on verb relations (this can be extended to other types of relations), we collect for each verb its subject(s), object(s), preposition(s) with arguments and auxiliary verb(s). We can now extract verb relations using a simple algorithm: We define a verb relation to be a verb together with its arguments (subject(s), object(s) and prepositional phrases) and consider only those relations to be of interest where at least the subject or the object is an NE. We filter out relations with only one argument.

2.3. Relation clustering

Relation clusters are generated by grouping relation instances based on their similarity.

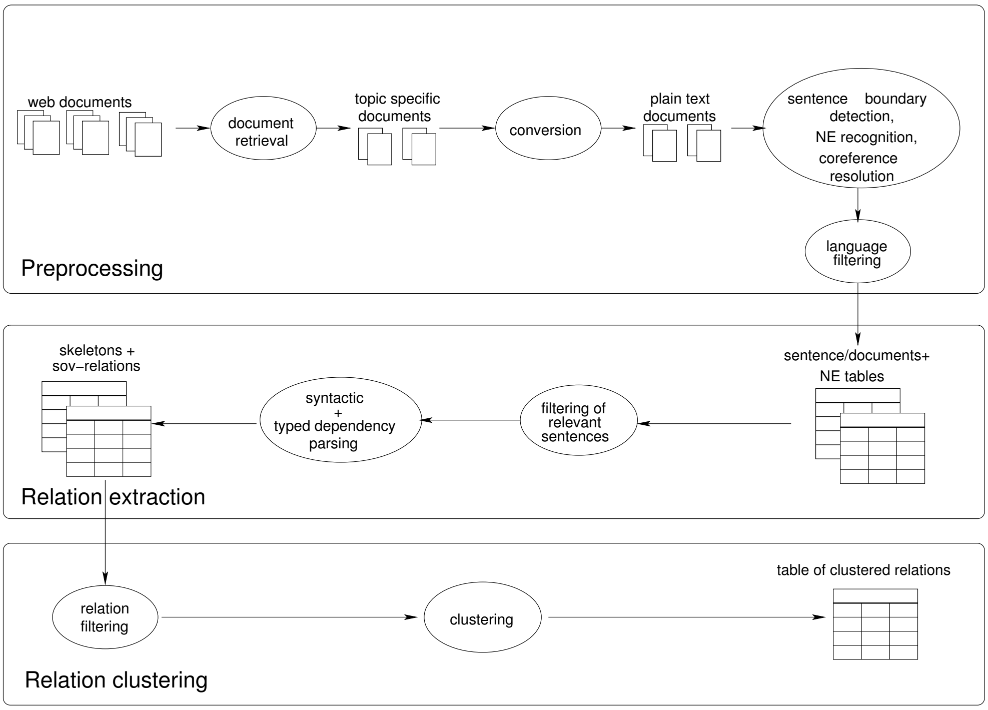
web documents
document retrieval
topic specific documents
conversion
plain text documents
sentence boundary detection, NE recognition, coreference resolution
language filtering
Preprocessing
sentence/documents+ NE tables
filtering of relevant sentences
syntactic + typed dependency parsing
skeletons + sov−relations
Relation extraction
relation filtering
clustering
table of clustered relations
Relation clustering

Figure 4: System architecture

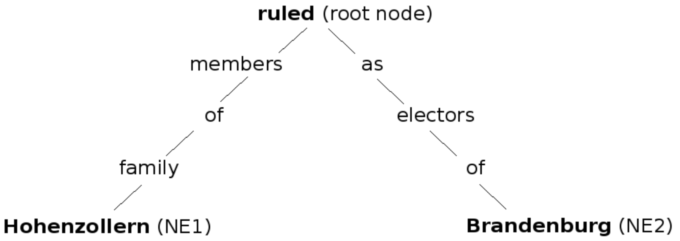

Figure 5: Skeleton for the NE pair "Hohenzollern" and "Brandenburg" in the sentence "Subsequent members of the Hohenzollern family ruled until 1918 in Berlin, first as electors of Brandenburg.“

The comparably large amount of data in the corpus requires the use of an efficient clustering algorithm. Standard ML clustering algorithms such as k-means and EM (as provided by the Weka toolbox (Witten and Frank, 2005)) have been tested for clustering the relations at hand but were not able to deal with the large number of features and instances required for an adequate representation of our dataset. We thus decided to use a scoring algorithm that compares a relation to other relations based on certain aspects and calculates a similarity score. If this similarity score exceeds a predefined threshold, two relations are grouped together.
Similarity is measured based on the output from the different preprocessing steps as well as lexical information from WordNet (WordNet, 2007):

- WordNet: WordNet information is used to determine if two verb infinitives match or if they are in the same synonym set.
- Parsing: The extracted dependency information is used to measure the token overlap of the two subjects and objects, respectively. We also compare the subject of the first relation with the object of the second relation and vice versa. In addition, we compare the auxiliary verbs, prepositions and preposition arguments found in the relation.
- NE recognition: The information from this step is used to count how many of the NEs occurring in the contexts, i.e., the sentences in which the two relations are found, match and whether the NE types of the subjects and objects, respectively, match.
- Coreference resolution: This type of information is used to compare the NE subject (or object) of one relation to strings that appear in the same coreference set as the subject (or object) of the second relation.

Manually analyzing a set of extracted relation instances, we defined weights for the different similarity measures and calculated a similarity score for each relation pair. We then defined a score threshold and clustered relations by putting two relations into the same cluster if their similarity score exceeded this threshold value.

3. Experiments and results

For our experiments, we built a test corpus of documents related to the topic “Berlin Hauptbahnhof” by sending queries describing the topic (e.g., “Berlin Hauptbahnhof”, “Berlin central station”) to Google and downloading the retrieved documents specifying English as the target language. After preprocessing these documents as described in 2.1., our corpus consisted of 55,255 sentences from 1,068 web pages, from which 10773 relations were automatically extracted and clustered.

3.1. Clustering

From the extracted relations, the system built 306 clusters of two or more instances, which were manually evaluated by two authors of this paper. 81 of our clusters contain two or more instances of exactly the same relation, mostly due to the same sentence appearing in several documents of the corpus. Of the remaining 225 clusters, 121 were marked as consistent, 35 as partly consistent, 69 as not consistent. We defined consistency based on the potential usefulness of a cluster to the user and identified three major types of potentially useful clusters:

- Relation paraphrases, e.g.,

  accused (Mr Moore, Disney, In letter)
  accused (Michael Moore, Walt Disney Company)

- Different instances of the same pattern, e.g.,

  operates (Delta, flights, from New York)
  offers (Lufthansa, flights, from DC)

- Relations about the same topic (NE), e.g.,

  rejected (Mr Blair, pressure, from Labour MPs)
  reiterated (Mr Blair, ideas, in speech, on March)
  created (Mr Blair, doctrine)
  ...

Of our 121 consistent clusters, 76 were classified as being of the type ’same pattern’, 27 as being of the type ’same topic’ and 18 as being of the type ’relation paraphrases’. As many of our clusters contain two instances only, we are planning to analyze whether some clusters should be merged and how this could be achieved.

3.2. Relation extraction

In order to evaluate the performance of the relation extraction component, we manually annotated 550 sentences of the test corpus by tagging all NEs and verbs and manually extracting potentially interesting verb relations. We define ’potentially interesting verb relation’ as a verb together with its arguments (i.e., subject, objects and PP arguments), where at least two of the arguments are NEs and at least one of them is the subject or an object. On the basis of this criterion, we found 15 potentially interesting verb relations. For the same sentences, the IDEX system extracted 27 relations, 11 of them corresponding to the manually extracted ones. This yields a recall value of 73% and a precision value of 41%.

There were two types of recall errors: First, errors in sentence boundary detection, mainly due to noisy input data (e.g., missing periods), which lead to parsing errors, and second, NER errors, i.e., NEs that were not recognised as such. Precision errors could mostly be traced back to the NER component (sequences of words were wrongly identified as NEs).

In the 550 manually annotated sentences, 1300 NEs were identified as NEs by the NER component. 402 NEs were recognised correctly by the NER, 588 wrongly and in 310 cases only parts of an NE were recognised. These 310 cases can be divided into three groups of errors. First, NEs recognised correctly, but labeled with the wrong NE type. Second, only parts of the NE were recognised correctly, e.g., “Tourismus Marketing GmbH” instead of “Berlin Tourismus Marketing GmbH”. Third, NEs containing additional words, such as “the” in “the Brandenburg Gate”.

To judge the usefulness of the extracted relations, we applied the following soft criterion: A relation is considered useful if it expresses the main information given by the sentence or clause, in which the relation was found. According to this criterion, six of the eleven relations could be considered useful. The remaining five relations lacked some relevant part of the sentence/clause (e.g., a crucial part of an NE, like the ’ICC’ in ’ICC Berlin’).

4. Possible enhancements

With only 15 manually extracted relations out of 550 sentences, we assume that our definition of ’potentially interesting relation’ is too strict, and that more interesting relations could be extracted by loosening the extraction criterion. To investigate on how the criterion could be loosened, we analysed all those sentences in the test corpus that contained at least two NEs in order to find out whether some interesting relations were lost by the definition and how the definition would have to be changed in order to detect these relations. The table in Figure 6 lists some suggestions of how this could be achieved, together with example relations and the number of additional relations that could be extracted from the 550 test sentences.

In addition, more interesting relations could be found with an NER component extended by more types, e.g., DATE and EVENT. Open domain NER may be useful in order to extract NEs of additional types. Also, other types of relations could be interesting, such as relations between coordinated NEs,

option	example	additional relations
extraction of relations, where the NE is not the complete subject, object or PP argument, but only part of it	*Co-operation with <ORG>M.A.X. 2001<\ORG>* <V>is<\V> clearly of benefit to <ORG>BTM<\ORG>.	25
extraction of relations with a complex VP	<ORG>BTM<\ORG> *<V>invited and or supported<\V>* more than 1,000 media representatives in <LOC>Berlin<\LOC>.	7
resolution of relative pronouns	The <ORG>Oxford Centre for Maritime Archaeology<\ORG> [...] *which* will <V>conduct<\V> a scientific symposium in <LOC>Berlin<\LOC>.	2
combination of several of the options mentioned above	<LOC>Berlin<\LOC> has *<V>developed to become<\V>* the *entertainment capital of <LOC>Germany<\LOC>*.	7

Figure 6: Table illustrating different options according to which the definition of ”potentially interesting relation” could be loosened. For each option, an example sentence from the test corpus is given, together with the number of relations that could be extracted additionally from the test corpus.

e.g., in a sentence like *The exhibition [...] shows <PER>Clemens Brentano<\PER>, <PER>Achim von Arnim<\PER> and <PER>Heinrich von Kleist<\PER>*, and between NEs occurring in the same (complex) argument, e.g., *<PER>Hanns Peter Nerger<\PER>, CEO of <ORG>Berlin Tourismus Marketing GmbH (BTM) <\ORG>, sums it up [...].*

5. Related work

Our work is related to previous work on domain-independent unsupervised relation extraction, in particular Sekine (2006), Shinyama and Sekine (2006) and Banko et al. (2007).

Sekine (2006) introduces On-demand information extraction, which aims at automatically identifying salient patterns and extracting relations based on these patterns. He retrieves relevant documents from a newspaper corpus based on a query and applies a POS tagger, a dependency analyzer and an extended NE tagger. Using the information from the taggers, he extracts patterns and applies paraphrase recognition to create sets of semantically similar patterns. Shinyama and Sekine (2006) apply NER, coreference resolution and parsing to a corpus of newspaper articles to extract two-place relations between NEs. The extracted relations are grouped into pattern tables of NE pairs expressing the same relation, e.g., hurricanes and their locations. Clustering is performed in two steps: they first cluster all documents and use this information to cluster the relations. However, only relations among the five most highly-weighted entities in a cluster are extracted and only the first ten sentences of each article are taken into account.

Banko et al. (2007) use a much larger corpus, namely 9 million web pages, to extract all relations between noun phrases. Due to the large amount of data, they apply POS tagging only. Their output consists of millions of relations, most of them being abstract assertions such as (executive, hired by, company) rather than concrete facts.

Our approach can be regarded as a combination of these approaches: Like Banko et al. (2007), we extract relations from noisy web documents rather than comparably homogeneous news articles. However, rather than extracting relations from millions of pages we reduce the size of our corpus beforehand using a query in order to be able to apply more linguistic preprocessing. Like Sekine (2006) and Shinyama and Sekine (2006), we concentrate on relations involving NEs, the assumption being that these relations are the potentially interesting ones. The relation clustering step allows us to group similar relations, which can, for example, be useful for the generation of answers in a Question Answering system.

6. Future work

Since many errors were due to the noisiness of the arbitrarily downloaded web documents, a more sophisticated filtering step for extracting relevant textual information from web sites before applying NE recognition, parsing, etc. is likely to improve the performance of the system.

The NER component plays a crucial role for the quality of the whole system, because the relation extraction component depends heavily on the NER quality, and thereby the NER quality influences also the results of the clustering process. A possible solution to improve NER in the IDEX System is to integrate a MetaNER component, combining the results of several NER components. Within the framework of the IDEX project a MetaNER component already has been developed (Heyl, to appear 2008), but not yet integrated into the prototype. The MetaNER component developed uses the results from three different NER systems. The output of each NER component is weighted depending on the component and if the sum of these values for a possible NE exceeds a certain threshold it is accepted as NE otherwise it is rejected.

The clustering step returns many clusters containing two instances only. A task for future work is to investigate, whether it is possible to build larger clusters, which are still meaningful. One way of enlarging cluster size is to extract more relations. This could be achieved by loosening the extraction criteria as described in section 4. Also, it would be interesting to see whether clusters could be merged. This would require a manual analysis of the created clusters.

Acknowledgement

The work presented here was partially supported by a research grant from the "Programm zur Förderung von Forschung, Innovationen und Technologien (ProFIT)" (FKZ: 10135984) and the European Regional Development Fund (ERDF).

7. References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI).*

Andrea Heyl. to appear 2008. Unsupervised relation extraction. Master's thesis, Saarland University.

Lc4j. 2007. Language categorization library for Java. http://www.olivo.net/software/lc4j/.

LingPipe. 2007. http://www.alias-i.com/lingpipe/.

Satoshi Sekine. 2006. On-demand information extraction. In *ACL*. The Association for Computer Linguistics.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proc. of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics.

Stanford Parser. 2007. http://nlp.stanford.edu/downloads/lex-parser.shtml.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, San Francisco, 2nd edition.

WordNet. 2007. http://wordnet.princeton.edu/.