Interactive Dynamic Information Extraction

Kathrin Eichler, Holmer Hemsen, Markus Löckelt, Günter Neumann, and Norbert Reithinger

Deutsches Forschungszentrum für Künstliche Intelligenz - DFKI,
66123 Saarbrücken and 10178 Berlin, Germany
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 for 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. In contrast to most of the current IE systems the IDEX system is domain-independent and tightly integrates a GUI for interactive exploration of the extraction space.

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 labor, such as specifying and implementing relation–specific extraction patterns manually or annotating large amounts of training corpora.

For example, in a hand–coded IE system a topic expert manually implements task–specific extraction rules on the basis of her manual analysis of a representative corpus. Note that in order to achieve a high performance the topic expert usually also has to have a very good expertise in Language Technology (LT) in order to specify the necessary mapping between natural language expressions and the domain–specific concepts. Of course, nowadays, there exists a number of available LT components, which can be used to preprocess relevant text documents and determine linguistic structure. However, this still requires a fine–grained and careful analysis of the mapping between these linguistic structures and the domain–knowledge.

This latter challenge is relaxed and partially automatized by means of corpus–based IE systems. Here, 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. Of course, also here existing LT core technology can be used to pre–compute linguistic structure. But still,

A. Dengel et al. (Eds.): LI 2008, LNAI 5243, pp. 54–61, 2008.

the heavy burden lies in a careful analysis of a very large corpus of documents with domain specific knowledge in order to support effective training of the ML engines.

One important issue for both approaches is, that the adaptation of an IE system to a new extraction task or domain have to be done offline, i.e., before the specific IE system is actually created. Consequently, current IE technology is highly statically and inflexible with respect to a timely adaptation of an IE system to new requirements in form of new topics.

1.1 Our Goal

The goal of our IE research is the development of a core IE technology to produce a new IE system automatically for a given topic on-demand and in interaction with an information analyst. 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 . This information request is used for an initial search in order to retrieve a relevant set of documents. This set of documents (i.e., the corpus) is then further passed over to Machine Learning algorithms which extract and collect (using the IE core components of IDEX) a set of tables of instances of possible relevant relations in an unsupervised way. These tables are presented to the user (who is assumed to be the topic–expert), who will investigate the identified relations further for his 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 navigate through the space of identified structure, e.g., in order to identify and specify new topic descriptions, which express his new attention triggered by novel relationships he was not aware beforehand.

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. In the next section, we further motivate the application potential and impact of the IDEX approach by an example application, before more technical details of the system are described.

1.2 Application Potential

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 tourist, 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 possible 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 for relevant text passages, in particular, if the names stand in relationship. Furthermore, because the found relationships are visualized using advanced graphical user interfaces, the user can select specific names and their associated relationships to other names, to the documents they occur in or she can search for paraphrases of sentences.

2 IDEX — System Overview

The next Fig. 1 shows the main components of the IDEX system. The system consists of two main parts: IdexExtractor, which is responsible for the information extraction, and IdexVisor, which realizes the graphical user interface.

Processing in IDEX is started by sending topic or domain relevant information in form of a search engine query to a web crawler by the user of the IDEX system. The set of documents retrieved are further processed by the IE core components, which perform Named Entity extraction, identification and clustering of interesting relations. All identified and extracted information units together with their textual and linguistic context are stored in different SQL tables, which are maintained by a standard SQL DB server. These DB tables are the input for the IdexVisor, which dynamically creates different visual representations of the data in the tables, in order to support flexible search and exploration of all entities by the user. Since all extracted information units are linked to each other and with their information sources, the user can simply bop around the different entities. For example, the user might decide to firstly investigate the extracted named entities, and then might jump to the positions of the original text sources in order to check the sentences in which interesting pairs of names appear. He can then decide to inspect the internal structure of the sentences in order to test, whether they belong to an interesting "hidden" semantic relationship. The user might then decide to specify a new topic in form of a more fine-grained search engine query in order to initiate a more focused web crawl.

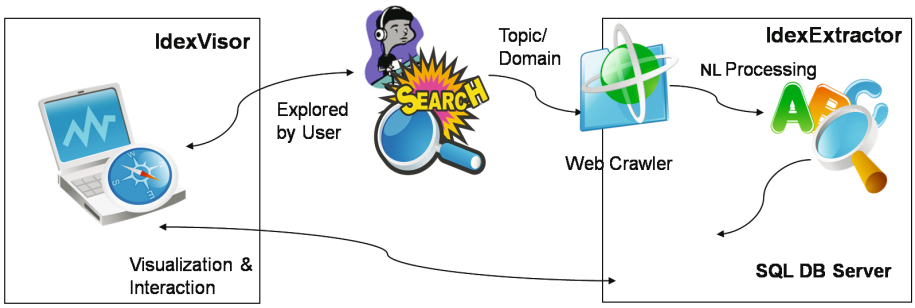

Fig. 1. Architecture of the IDEX System

2.1 IdexExtractor — Unsupervised Relation Extraction

Preprocessing After specifying the topic the documents (HTML and PDF) are automatically crawled from the web using the Google search engine and converted into plain text files. We apply LingPipe [1] 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).

Relation extraction. 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 [2] 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. In the second step, information based on dependency types is extracted for the potentially relevant sentences. We focus on verb relations and collect for each verb its subject(s), object(s), preposition(s) with arguments and auxiliary verb(s). We consider only those relations to be of interest where at least the subject or the object is an NE. Relations with only one argument are filtered out.

Relation clustering. Relation clusters are generated by grouping relation instances based on their similarity. Similarity is measured based on the output from the different preprocessing steps as well as lexical information. WordNet [3] information is used to determine if two verb infinitives match or if they are in the same synonym set. The information returned from the dependency parser is used to measure the token overlap of the two subjects and objects, respectively. In addition, we compare the auxiliary verbs, prepositions and preposition arguments found in the relation. We count how many of the NEs match in the sentences in which the two relations are found, and whether the NE types of the subjects and objects match. 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.

2.2 IdexVisor — Interactive Exploration of the Extraction Space

Using the IdexVisor-Frontend, a user can access different visualizations of the extracted data and he can navigate through the data space in a flexible and dynamic manner. IdexVisor is a platform independent application. It can be configured dynamically with respect to the underlying structure of the data base model used by IdexExtractor using an XML–based declarative configuration.

The Frontend is placed between the user and a MySQL DB server. IdexVisor follows the *Model-View-Controller*-Approach (MVC), as is illustrated in Fig. 2. The results of IdexExtractor are available in form of a number of tables. Each table represents one aspect of the extracted data, e.g., a table contains

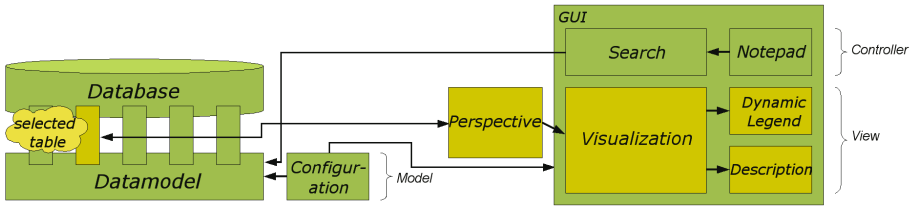

Fig. 2. Architecture of IdexVisor

all sentences of all documents with information about the textual content, the document link, the language and a pointer to the topic the sentence belongs to. The configuration of the Frontend specifies meta–information about the part of the data that has to be visualized. With help of the *View*, the selected data is visualized together with additional information, e.g., a dynamic help comment or a caption. Using the *Controller* a user can navigate through the selected data, he can search for specific information or he can specify a queries for a search engine. The visualization offers a set of *perspectives* that offer different viewpoints on subsets of the data that are accessible via separate tabs on the GUI. For example, one perspective shows named entities, and their inferred type (person, geographical location, or organization), clustered according to the documents they occur in; another clusters entities according to whether they are syntactical arguments (subject, object, preposition etc.) to the same predicate. The perspectives are intended to be used in combination to find the answers to questions; query terms can therefore be transferred from one perspective to another.

3 Experiments

We have performed two different experiments in order to test the two subsystems. In the first experiment we evaluated the quantity and quality of the output of IdexExtractor. In the second experiment, a number of test users measured the performance of IdexVisor.

3.1 Experiments with IdexExtractor

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, our corpus consisted of 55,255 sentences from 1,068 web pages, from which 10773 relations were automatically extracted and clustered.

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 (i.e., all instances in the cluster express a similar relation), 35 as partly consistent (i.e., more than half of the instances in the cluster express a similar relation), 69 as not useful. The clusters marked as consistent can be grouped into three major types 1) relation paraphrases 2) different instances of the same pattern 3) relations about the same topic (NE). 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.

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) (see [4] for details). 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').

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, for example, by extraction of relations where the NE is not the complete subject, object or PP argument, or by extraction of relations with a complex VP. Further details are presented in [4].

3.2 Experiments with IdexVisor

Seven users (average age 33; 4 males/3 females) evaluated the IdexVisor-Frontend. After an introduction of the functionality of the system and a demonstration of a complex search query, the users tried to answer the following four corpus–related questions via interaction with the system:

1. *Find out information about a person"Murase". The complete name, whether the person owns a company, and if so, what is the name of it.*
2. *Find one or more documents in which Hartmut Mehdorn, the CEO of the Deutsche Bundesbahn, and Wolfgang Tiefensee, the Minister of Transport occur together.*
3. *Who built the Reichstag and when? During your search also use the synonym perspective of the system in order to find alternative predicates.*
4. *How often is Angela Merkel mentioned in the corpus?*

The users were then asked about different aspects of the interaction of the system. For each question they could give a real number as answer (cf. Fig. 3) and a short text, where they could describe what they like, what they missed, and could suggest possible improvements. Here are the results of the verbal answers:

Overall usability: All users were able to answer the questions, but with different degree of difficulties in the interaction. Users stated that (a) switching between perspectives was perceived cumbersome, (b) the benefits not obvious, and (c) the possibility was not salient to them, although it was explained in the short introduction to the system. Different types of search queries were not recognized. Parts of the user interface were overlooked or actually not recognized. It was also said, that the short introduction time (10 minutes) was not enough to complete understood the system. The search speed was judge generally as "fast".

Possible Improvements: The synonyms should not be represented as a separate perspective, but should be integrated automatically with the search. The clustering according to semantic similarity should be improved. The major critical point for the representation of the relations was that the text source was only shown for some nodes and not all.

Question	**Possible Answers**	∅
How did you like the introduction ?	1=useless/5=helpful	4,42
How useful is the system?	1=useless/5=helpful	4,14
Do you think you might use such a system in your daily work?	1=no/5=yes	4,14
How do you judge the computed information?	1=useless/5=very informative	3,71
How do you judge the speed of the system?	1=very slow/5=very fast	4,42
How do you judge the usability of the system?	1=very laborious/5=very comfortable	3,42
Is the graphical representation of the results useful?	1=totally not/5=very useful	3,57
Is the graphical representation appealing?	1=totally not/5=very appealing	3,71
Is the navigation useful in the system ?	1=totally not/5=very useful	3,57
Is the navigation intuitive in the system?	1=totally not/5=very intuitive	3,57
Did you have any problems using the system?	1=heavy/5=no difficulties	4,28

Fig. 3. Results of the evaluation of IdexVisor

4 Related Work

The tight coupling of unsupervised IE and interactive information visualization and navigation is, to best of our knowledge, novel. The work on IDEXVISOR has been influenced by work on general interactive information visualization techniques, such as [5]. Our work on relation extraction is related to previous work on domain-independent unsupervised relation extraction, in particular Shinyama and Sekine [6] and Banko et al. [7]. Shinyama and Sekine [6] 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. [7] 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 the two approaches: Like Banko et al. [7], 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. Unlike Banko et al. [7], we concentrate on relations involving NEs, the assumption being that these relations are the potentially interesting ones.

References

1. LingPipe, `http://www.alias-i.com/lingpipe/`
2. Stanford Parser, `http://nlp.stanford.edu/downloads/lex-parser.shtml`
3. WordNet, `http://wordnet.princeton.edu/`
4. Eichler, K., Hemsen, H., Neumann, G.: Unsupervised relation extraction from web documents. In: Proceedings of the 6th edition of the Language Resources and Evaluation Conference (LREC 2008), Marrakech, Morocco, May 28–30 (2008)
5. Heer, J., Card, S.K., Landay, J.A.: prefuse: a toolkit for interactive information visualization. In: CHI 2005, Human Factors in Computing Systems (2005)
6. Shinyama, Y., Sekine, S.: 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, Association for Computational Linguistics, pp. 304–311 (2006)
7. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: Proc. of the International Joint Conference on Artificial Intelligence (IJCAI) (2007)