ENRYCHER – SERVICE ORIENTED TEXT ENRICHMENT

Tadej Štajner, Delia Rusu, Lorand Dali, Blaž Fortuna, Dunja Mladenić, Marko Grobelnik
Department of Knowledge Technologies
Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773419; fax: +386 1 4251038
e-mail: tadej.stajner@ijs.si

ABSTRACT

This paper describes a natural language processing and information extraction framework and illustrates several use cases where the service-oriented approach has proven to be useful. We also describe an abstract extensible schema model for representing document enrichments.

1 INTRODUCTION

In our experience, many knowledge extraction scenarios generally consist of multiple steps, starting with natural language processing, which are in turn used in higher level annotations, either as entities or document-level annotations. This in turn yields a rather complex dependency scheme between separate components. Such complexity growth is a common scenario in general information systems development. Therefore, we decided to mitigate this by applying a service-oriented approach to integration of a knowledge extraction component stack. The motivation behind Enrycher[17] is to have a single web service endpoint that could perform several of these steps, which we refer to as 'enrichments', without requiring the user to bother with setting up pre-processing infrastructure himself.

The following chapters will describe the specific components, integration details and some of the use cases that motivated this experiment of integration.

2 RELATED WORK

There are various existing systems and tools that tackle either named entity extraction and resolution, identification of facts, document summarization. The OpenCalais system [15], for example, creates semantic metadata for user submitted documents. This metadata is in the form of named entities, facts and events. In the case of our system, named entities and facts represent the starting point; we identify named entities within the document, determine the subject - verb - object triplets, and refine them by applying co-reference resolution, anaphora resolution and semantic entity resolution. As opposed to OpenCalais, we continue the pipeline to extract assertions from text, which represent newly identified relationshops, present in text. This process enables the construction of a semantic description of the document in the form of a semantic directed graph where the nodes are the subject and object triplet elements, and the link between a pair of entities is determined by the verb (predicate triplet element). The initial document, its associated triplets and semantic graph are then employed to automatically generate a document summary. The resulting triplets are then in turn used to construct a semantic graph, an effective and concise representation of document content [12].

3 ARCHITECTURE

The process consists of several phases, each depending on the output of the previous one. The dependencies between components can be illustrated by the following chart (Figure 1):

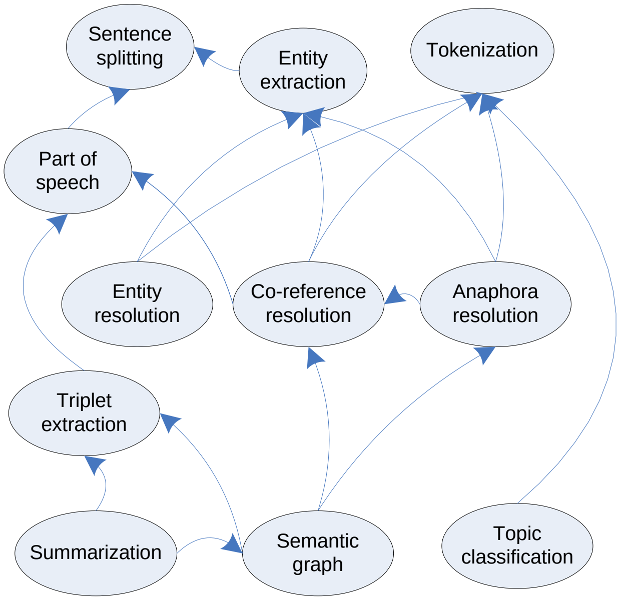

Figure 1: Component dependency graph

3.1 LANGUAGE-LEVEL PROCESSING

While language-level features are usually not explicitly stated as a requirement in most use cases, they are instrumental to most of the further enrichments that are required in those use cases:

- Sentence splitting
- Tokenization
- Part of speech tagging
- Entity extraction

3.2 ENTITY-LEVEL PROCESSING

Whereas the language-level processing step identified possible entities, the purpose of this phase is to consolidate the identified entities. This is done with anaphora resolution, where pronoun mentions are merged with literal mentions, co-reference resolution that merges similar literal mentions and entity resolution, which links the in-text entities to ontology concepts.

3.2.1 NAMED ENTITY EXTRACTION

We gather named entities in text using two distinct approaches to named entity extraction, a pattern-based one [9] and a supervised one [10].

3.2.1 ANAPHORA RESOLUTION

Anaphora resolution is performed for a subset of pronouns: {I, he, she, it, they}, and their objective, reflexive and possessive forms, as well as the relative pronoun who. A search is done throughout the document for possible candidates (named entities) to replace these pronouns. The candidates receive scores, based on a series of antecedent indicators (or preferences): givenness, lexical reiteration, referential distance, indicating verbs and collocation pattern preference [1].

3.2.2 CO-REFERENCE RESOLUTION

Co-reference resolution is achieved through heuristics that consolidate named entities, using text analysis and matching methods. We match entities where one surface form is completely included in the other, one sufrace form is the abbreviation of the other, or there is a combination of the two situations described [1].

3.2.3 SEMANTIC ENTITY RESOLUTION

Rather than just extracting information from text itself, the motivation behind entity resolution is to integrate text with an ontology. This consists of matching previously extracted named entities to ontology concepts. Since named entities are often ambiguous, especially in multi-domain ontologies, such as DBpedia [13], we have to employ sophisticated methods to determine the correct corresponding semantic concept of a named entity. The underlying algorithm uses ontology entity descriptions as well as the ontology relationship structure to determine which are the most likely meanings of the named entities, appearing in the input text. Because the approach is collective, it does not treat distinct entity resolution decisions as independent. This means that it can successfully exploit relational similarity between ontology entities, which means that entities, which are more related to each other, tend to appear more ofter together. This is explored in further detail in [11], with concrete implementation details in [6].

3.3 ENTITY GRAPH PROCESSING

3.3.1 TRIPLET EXTRACTION

The triplet is a semantic structure composed of a subject, a verb and an object. This structure is meant to capture the meaning of a sentence. We try to extract one or more triplets from each sentence independently. Two approaches to triplet extraction have been tried, both of which take as input a sentence with tokens tagged with their part of speech.

In the first approach the sentence is parsed, and then the triplets are extracted based on the shape of the parse tree obtained. The rules of triplet extraction from a parse tree are explained in detail in [5].

In order to avoid the performance bottleneck introduced by deep parsing, we tried another approach where instead of parsing, we only do noun phrase chunking on the input sentence. The result of chunking is a sequence of tags on which pattern matching rules are applied in order to find the triplets which must be extracted. This pattern matching rules are similar to regular expressions applied on text. The difference is that as opposed to regular expressions which have as the processing unit a character, the triplet extraction rules recognise the tags as the smallest units which can be matched.

The second approach brings an important speedup to the triplet extraction process. However, due to the sequential structure of the chunked sentence, it loses some of the representational power when compared to the richer structure of a parse tree. This is why it is more difficult, if not impossible, to find some of the triplets in a chunked sentence than finding them in a parsed sentence. Another advantage of the chunked approach is that the pattern matching rules are easier to understand and extend.

3.4 DOCUMENT-LEVEL PROCESSING

While the language-level processing operates on the token and phrase domain and the entity-level processing operates on the in-text entities and concepts, the document-level processing uses the preceding enrichments to annotate the document as a whole.

3.4.1 SEMANTIC GRAPH VISUALIZATION

The semantic representation of text is achieved through linking triplet elements together, where the nodes are represented by the subject and object elements, and the relationship between them is linked with the corresponding verb. The yielded graph is a directed one, from the subject element to the object one.

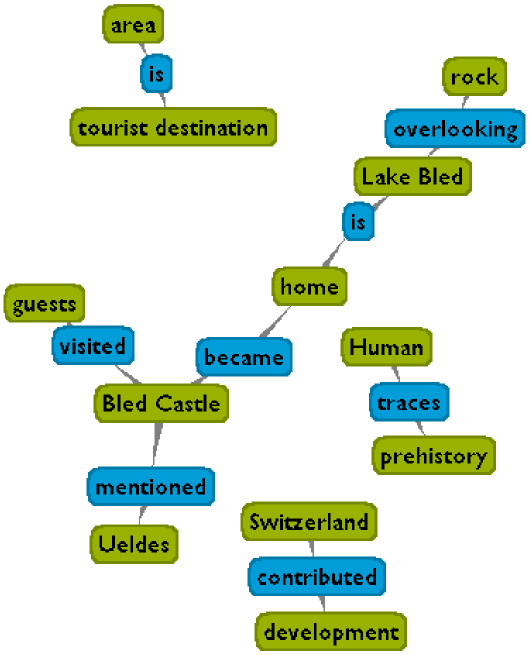

Figure 2: Example of a semantic graph visualization: Wikipedia article on Bled

Thus we can represent plain-text in a more compact manner that enables visual analysis, highlighting the most important concepts and the relations among them. An example is illustrated in Figure 2.

3.4.2 TAXONOMY CATEGORIZATION

A common use case in working with documents is classifying them in categories. This component annotates the input text with a hierarchical classifier which chooses relevant categories based on word and phrase similarity [14]. The current on-line implementation uses the Open Directory as an example of a taxonomy.

3.4.3 CONTENT SUMMARIZATION

The document's semantic graph is a starting point for automatically generating the document summary. The model for summary generation is obtained by machine learning, where the features are extracted from the semantic graph structure and content [1].

3.5 MODEL SCHEMA

The schema that is used in the inter-service communication is abstracted to the point that it is able to represent:

- **Document-wide metadata**: identifier, document-wide semantic attributes (e.g. categories, summary),
- **Text**: sentences, tokens, part of speech tags,
- **Annotations**: entities and assertion nodes, identified in the article with all identified instances, possibly also with semantic attributes (e.g. named entities, semantic entities)
- **Assertions**: identified <*subject, predicate, object*> triplets, where subjects, predicates and object themselves are annotations.)

4 USAGE

The system abstracts the setup and workflow from the user by exposing only a single web service endpoint, which in turn pipelines the request thorough other web services. All communication is done with REST-like XML-over-HTTP requests.

5 USE CASES

5.1. VISUAL ANALYTICS

Visual analysis of documents based on the semantic representation of text in the form of a semantic graph can aid data mining tasks, such as exploratory data analysis, data description and summarization. Users can thus get an overview of the data, without the need to entirely read it. This kind of concept overview offers straightforward data visualization by listing the main facts (the triplets), linking them in a way that is meaningful for the user (the semantic graph), as well as providing a document summary. [4].

5.2 SEMANTIC INTEGRATION OF TEXT AND ONTOLOGIES

An important part of information systems integration is providing interoperability of data. This is a major issue when dealing with plain text, because it is inherently unstructured. On the other hand, one of the most pragmatic approaches is representing knowledge in a common ontology. Therefore, we designed our system to not only identify and consolidate named entities in text, but uses the semantic entity resolution component to match it with ontology concepts, which enables us to represent nodes in the graph as semantic concepts.

5.3 QUESTION ANSWERING

Document enrichment techniques such as triplet extraction and semantic graphs have been applied to build a question answering system [3]. The use case is that the answer to a natural language question is searched in a collection of documents from which triplets have been previously extracted. Triplets, possibly incomplete, are also extracted from the question, and they are matched against the triplets extracted from the documents to find the answers.

5.4 STORY LINK DETECTION

A task, related to news mining and analysis is story link detection [7], where the objective is to identify links between distinct articles that form a coherent story. [2] shows that enriching the text with entity extraction and resolution improves story link detection performance. This indicates that such enrichment on documents may also be beneficial for other topic detection and tracking or semantic search tasks.

6 DISCUSSION AND FUTURE WORK

A use case for Enrycher in a related domain of computational linguistics is evaluating local discourse coherence of text. This is an intrinsic measure that indicates readability of text. Since it is automatic, it is also convenient for large-scale evaluation of automatically generated text. The concrete method is based on detecting rough shifts in entity mentions and short entity topics as indicators of poor coherence. As Enrycher supplies grammar roles and entities in triplets, we can match them to the sentences they have been extracted from and evaluate discourse coherence.

Another interesting research area that we are currently tackling is extracting knowledge from large-scale document collections, such as news corpora, where we are exploring possible usability and visualization improvements. Since we extract triplets and possibly resolve their nodes to semantic concepts, we can create new ontologies from corpora of text automatically. Since we are able to do semantic entity resolution, we can also perform alignment of newly extracted ontologies with other ontologies.

As of writing, we are developing additional applications that use Enrycher at their cores. One such example is a mobile RSS news reader, which leverages Enrycher to perform text summarization on news items to make them more suitable to consume on a screen space constrained mobile device.

7 CONCLUSION

We show that Enrycher offers a user-friendly way to qualitatively enhance text from unstructured documents to semi-structured graphs with additional annotations. Since the system offers a full knowledge extraction stack, it makes the system simpler to use than having the user to implement and configure several processing steps that are usually required in knowledge extraction tasks. We described various use cases in both research and applied tasks which we were able to solve with the use of Enrycher as infrastructure.

8 ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the IST Programme of the EC under ACTIVE (IST-2008-215040) and PASCAL2 (IST-NoE-216886).

References

[1] D. Rusu, B. Fortuna, M. Grobelnik and D. Mladenić: Semantic Graphs Derived From Triplets With Application. *In Document Summarization. Informatica Journal,* 2009

[2] T. Štajner, M. Grobelnik: Story Link Detection with Entity Resolution. In Proceedings of Semantic Search Workshop at WWW2009, Madrid, Spain, 2009

[3] L. Dali, D. Rusu, B. Fortuna, D. Mladenić, M. Grobelnik: Question Answering Based on Semantic Graphs. In Proceedings of *Semantic Search* at WWW2009, Madrid, Spain, 2009

[4] D. Rusu, B. Fortuna, D. Mladenić, M. Grobelnik and R. Sipoš, Proceedigns of *Visual Analysis of Documents with Semantic Graphs* Workshop *VAKD '09 at KDD-09*

[5] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, D. Mladenić. Triplet Extraction from Sentences. Ljubljana: 2007. In *Proceedings of the 10th International Multiconference "Information Society - IS 2007".* Vol. A, pp. 218 - 222.

[6] T. Štajner, From unstructured to linked data: entity extraction and disambiguation by collective similarity maximization. In *Proceedings of Identity and reference in web-based knowledge representation Workshop* at IJCAI 2009

[7] J. Allan. Introduction to Topic Detection and Tracking. *Kluwer Academic Publishers, Massachusetts, 2002, pp. 1–16.*

[8] J.J. Thomas and K.A. Cook. A Visual Analytics Agenda. *IEEE Comput. Graph. Appl.* 26, 1 (Jan. 2006), 10-13.

[9] H. Cunningham, GATE, a general architecture for text engineering, *Computers and the Humanities, 2002*

[10] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.*

[11] X. Li, P. Morie, and D. Roth, Semantic integration in text: From ambiguous names to identifiable entities,*" AI Magazine. Special Issue on Semantic Integration, vol. 26, no. 1, pp. 45-58, 2005.*

[12] I. Herman, G Melançon, M.S. Marshal: Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics, 2000.*

[13] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, Dbpedia: A nucleus for a web of open data*, Lecture Notes in Computer Science, vol. 4825, p. 722, 2007.*

[14] M. Grobelnik, D. Mladenić. Simple classification into large topic ontology of Web documents. In *Proceedings of the 27th International Conference on Information Tech-nology Interfaces, 20-24 June, Cavtat, Croatia, 2005.*

[15] OpenCalais, *http://www.opencalais.com/*

[16] R. Barzilay, M. Lapata. Modeling Local Coherence: An Entity-Based Approach. In *Computational Linguistics,* Vol. 34, No. 1, Pages 1-34, *2008*

[17] Enrycher, http://enrycher.ijs.si