A Service Oriented Framework for Natural Language Text Enrichment

Tadej Štajner, Delia Rusu, Lorand Dali, Blaž Fortuna, Dunja Mladenić and 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

Keywords: language processing, information extraction, entity resolution, automatic document categorization

Received: November 18, 2009

This paper describes a text enrichment framework and the corresponding document representation model that integrates natural language processing, information extraction, entity resolution, automatic document categorization and summarization. We also describe the implementation of the framework and give several illustrative use cases where the service-oriented approach has proven to be useful.

Povzetek: Opisan je okvir za obogatitev naravnega besedila.

1 Introduction

Integration and sharing of data across different data sources is the basis for an intelligent and efficient access to multiple heterogeneous resources. Since a lot of knowledge is available in plain text rather than a more explicit format, an interesting subset of this challenge is integrating texts with structured and semi-structured resources, such as ontologies. However, this also involves dealing with natural language, which is an error-prone process.

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 users to bother with setting up pre-processing infrastructure themselves.

The next sections will describe the specific components, integration details and some of the use cases that motivated this experiment of integration. Section 2 describes related ideas and their implementations and positions our model within their framework. Section 3 describes the overall architecture, the model and individual components. Section 4 focuses on the applications and use cases of the framework, continuing to the conclusion in section 5 and continueing to the outline of the current and future work that is going on related to the Enrycher framework and system.

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 relationships expressed 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. 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].

The enrichment where we provide a set of triplets in the form subject, predicate, object can be also described as a form of information extraction, since we are extracting structure from inherently unstructured data. Information extraction approaches can be distinguished in several ways. The first distinction comes from the way the background knowledge is being used.

Open information extraction approaches do not assume anything about the relations they might encounter and do not require training data. Their main characteristic is that they work accross a wide array of domains, it is a trait which we wanted to keep in Enrycher. Another prominent example in this area is TextRunner [18],

Supervised information extraction approaches, such as WebKB [19], are suitable for domain-specific text, since they require training data for each relationship that we wish to extract, although more recent approaches tend to focus on approches which require less human supervision.

Semi-supervised information extraction approaches are most often implemented via starting with a bootstrap training set and learning new patterns along the way, for example, the recent ReadTheWeb approach [20]. While it still requires the concrete relations to be specified via the training set, it requires much less human supervision to improve the extraction by learning. Recently, hybrid approaches combining open and bootstrapped information extraction have started to appear [21]. Since the overall tendency in the area of information extraction is reducing the human involvement within the extraction process, we positioned Enrycher's information extraction capabilities to be closer to an open information extraction system which would use not only literal pattern matching for information extraction but also extracting patterns with the help of part-of-speech information. This enables us to extract statements that would be otherwise missed by literal patterns and to provide us with a starting point for language independence.

Another difference within information extraction approaches is the treatment of extracted terms, such as named entities:

Well-defined entities and relationships are a property of the knowledge model which asserts that a single term has only a single meaning. In that case, we refer to terms as entities. We achieve this property by performing entity resolution. However, this is not always necessary and depends on the desired use of the knowledge base. For instance, if a knowledge base is exposed as an information retrieval engine, such as TextRunner [18], StatSnowball [21] and Read The Web [20], ambiguity is not a significant issue. However, if we wish to use the knowledge base to perform logical inference or any other task with similar demands, entity and relationship resolution is necessary, as demonstrated in SOFIE [22]. Enrycher uses a combination of both - named entity extraction may detect more terms that the entity resolution can then resolve into concrete entities, allowing for both to co-exist.

3 Architecture

The process underlying the proposed framework consists of several phases, each depending on the output of the previous one.

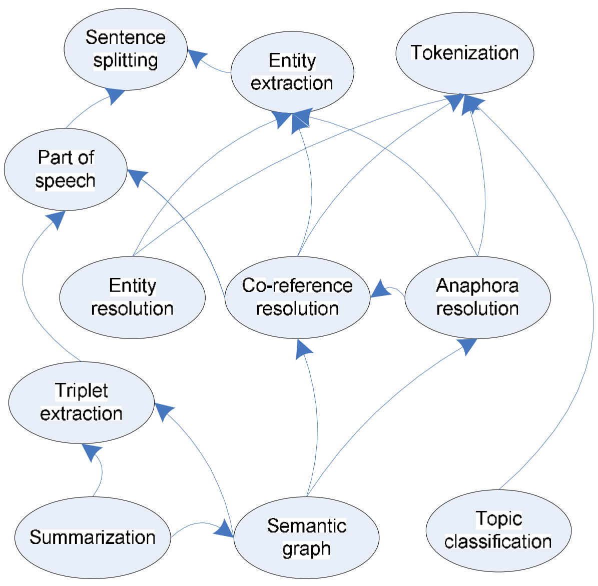

Figure 1: Component dependency graph.

The dependencies between components can be illustrated by the chart in Fig. 1. The architecture can be roughly split into four phases which correspond to four different parts of the proposed document enrichment model.

3.1 Model schema

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

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

The first part of the enrichment model is the linguistic description of the text containing the output of initial natural language. The data structures at this level are blocks (i.e. paragraphs, titles), which are then split into sentences and later into individual words.The second part of the enrichment model is the annotation section, containing information on individual objects that have been detected in the document one way or another, for instance, named entities and other semantic graph nodes. The annotations themselves contain a list of their instantiations within the document and a set of associated semantic attributes, meant for describing them and linking them to an existing ontology

The third part of the enrichment model is the assertions section, containing the triplets that construct the semantic graph. This represents the individual information fragments that were extracted from the plain text and form the basis for new knowledge.

Furthermore, each document contains a document metadata section storing attributes that apply to the document as a whole, such as categories, descriptive keywords and summaries. The next sections will individually describe each of the parts and outline the approaches that have been employed to implement the actual software that provides data for the enrichment model.

3.2 Language-level processing

While language-level features usually aren't 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

This is also the layer that is the most language-dependent, providing an abstraction to the other layers. Ideally, the tools above this layer should be reasonably lanugage-agnostic, although some concrete implementation approaches might not make this entirely possible. For instance, semantic entity resolution requires a background knowledge base to map from concrete phrases to concepts, requiring the phrases to be in the language that is used. On the other hand, if the triplet extraction layer works only on top of part-of-speech, such as the one implemented within Enrycher, it is reasonably language-agnostic, since it operates at a higher level than matching literal strings.

3.3 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. Since entity extraction is often handled with several domain-specific extractors, the purpose of this layer is to allow multiple extraction mechanisms and consolidate their output into a coherent set of entities and if possible, linking them to ontology concepts.

3.3.1 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]. This step is done to gather as much annotations as possible to have a rich node set for the semantic graph.

3.3.2 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].

3.3.3 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 in [1]. This is useful for reducing the size of the graph, as it enables us to merge many nodes into one, such as merging mentions, such as *"Mr. Norris"* and *"Chuck Norris"* into a single entity.

3.3.4 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, it 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 [23] and [6], where the improvement in entity resolution quality with using heterogeneous graph structure is demonstrated.

3.4 Entity graph processing

3.4.1 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.

Triplet extraction by deep parsing was the first approach that was tried. In this approach the sentence is

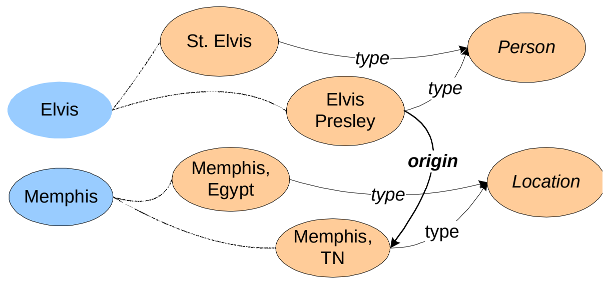

Figure 2: An example of an entity resolution scenario, where the blue nodes represent in-document extracted entities and the pink nodes represent the ontology concepts to which we align our document.

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]. *Triplet extraction by noun phrase chunking and pattern matching* was employed 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. An example:

```
   Triplet <-
{:subject:<NP>}(<PP><NP>)*
{:verb:<VP>(|<JJ.*>)*}
{:object:<NP>+}
```

Figure 2: Triplet extraction pattern

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, such as in Fig. 2, are easier to understand and extend.

3.5 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.

3.5.1 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.

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 Fig. 3.

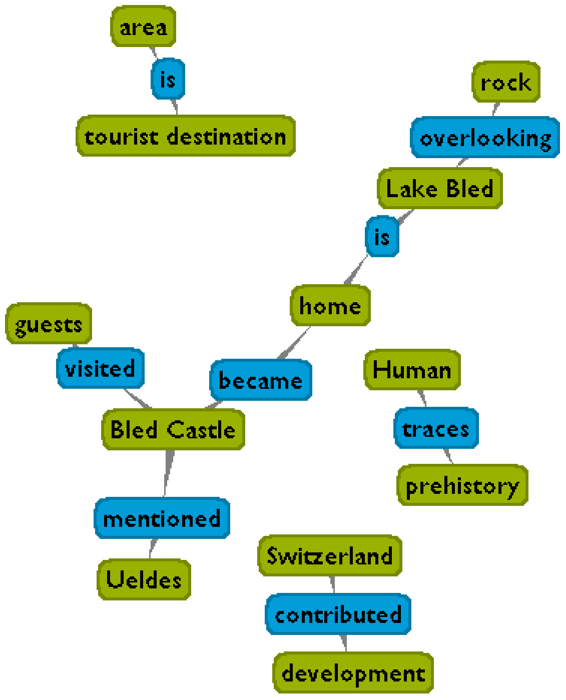

Figure 3: Example of a semantic graph visualization: Wikipedia article on Bled

3.5.2 Taxonomy categorization

A common use case in working with documents is classifying them in categories. This component annotates the input text with a hierarchical centroid 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. The same component also provides descriptive keywords, generated from segments of the category names. For instance:

```
Categories:
•  Top/Reference/Knowledge_Managemen
   t/Knowledge_Representation
•  Top/Reference/Knowledge_Managemen
   t
•  Top/Reference/Knowledge_Managemen
   t/Knowledge_Representation/Ontolo
   gies
•  Top/Reference/Knowledge_Managemen
   t/Knowledge_Discovery

Keywords:
•  Knowledge Management,
•  Knowledge Representation,
•  Ontologies
```

In this way it is possible to provide enrichments that do not necessarily correspond to an explicit mention of an entity but rather to the document as a whole, such as document classification labels.

3.5.3 Content summarization

The document's semantic graph is a starting point for automatically generating the document summary. This summarization approach is based on the idea that more imporant entities are positioned more centrally in the semantic graph which represent the central topic of the document. Therefore, if we reduce the graph in less dense areas, we are removing less important sentences.The model for summary generation is obtained by machine learning, where the features are extracted from the semantic graph structure and content [1].

3.5.4 Other options

Although not directly implemented in the Enrycher system, the document model is sufficient to represent also other enrichments. For instance, other document-wide enrichments can be the automatically detected language of the document, the detected sentiment of the document, references to possibly related documents and similar. The assertions are also able to represent RDF assertions, making it amenable to Semantic Web-oriented applications and knowledge acquisition.

4 Use cases

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.

4.1 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].

4.2 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 use the semantic entity resolution component to match it with ontology concepts, which enables us to represent nodes in the graph as semantic concepts. This can be an important aid in constructing ontologies from textual data.

4.3 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.

4.4 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.

5 Conclusion

The main contribution of this paper is an integrrated framework for enriching textual data based on natural language information extraction to include more structure and semantics. We implemented the proposed framework in the system, named Enrycher, which 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 text enrichment 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.

Furthermore, such systems can be used as infrastructure for knowledge acquisition. Recently, much emphasis in both the academic and industrial communities has been given to the Linked Open Data [24], proposing common RDF-based vocabularies for databases to allow easy integration. The consequence of this for knowledge extraction engines is that it is desired to have a knowledge representation rich enough to be expressed in RDF, that inevitably means resolving phrases to entities and verbs to well-defined relations, identifiable by Unique Resource Identifiers. This push on stricter data representation also brings along more difficult knowledge acquisition. Weaker representations can tolerate wrong assertions more easily, since they make less assumptions about their truth value. On the other hand, ontologies assume that the statements are true in their model, this can lead to erratic behaviour of applications, depending on those assumptions. We therefore expect to see further work on constrained ontology population from extracted information. On the other hand, our paper barely touches the possibilities that could be employed by using globally identified data approaches, opening way for better data integration, visualization and using annotated documents to enable semantic search. We expect that the proposed semantic article enrichment method will yield even more improvement on tasks that depend on the added semantic information, such as document summarization, triple extraction and recommendation systems.

6 Current 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've 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.

Our ongoing work is on 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.

Other future work will focus on using Enrycher as an automated approach to knowledge acquisition which will be able to use the obtained knowledge to improve its output quality. Another path of possible improvement is to test the language-independence aspect of higher-level processing stages, such as anaphora and coreference resolution and also semantic entity resolution and demonstrate whether this sort of framework is able to support multiple different languages withing its processing pipeline, an important requirement for using data sourced from the Web.

Acknowledgement

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. *Semantic Search at WWW2009, Madrid, Spain*, 2009

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

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

[5] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, D. Mladenić: Triplet extraction from sentences, Proceedings of the 10th International Multiconference on Information Society, *SiKDD 2007subconference.*

[6] T. Štajner: From unstructured to linked data: entity extraction and disambiguation by collective similarity maximization. In *Identity and reference in web-based knowledge representation 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: 27th International Conference on Information Technology 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, pp. 1-34, *2008*

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

[18] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. *Proc. of IJCAI, 2007.*

[19] R. Ghani, R. Jones, D. Mladenic, K. Nigam, and S. Slattery. Data mining on symbolic knowledge extracted from the web. *In Workshop on Text Mining at the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. 56.* Citeseer, 2000.

[20] Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information extraction. *In Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010),* 2010.

[21] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.R. Wen. StatSnowball: a statistical approach to extracting entity relationships. *In Proceedings of the 18th International Conference on World Wide Web*, pp. 101-110. ACM New York, NY, USA, 2009.

[22] F.M. Suchanek, M. Sozio, and G. Weikum. SOFIE: a self-organizing framework for information extraction. *In Proceedings of the 18th International Conference on World Wide Web*, pp. 631- 640. ACM, 2009.

[23] Tadej Štajner, Dunja Mladenić: Entity Resolution in Texts Using Statistical Learning and Ontologies, *In Proceedings of the 4th Asian Semantic Web Conference*, pp. 91-104, 2009

[24] C. Bizer, T. Heath, and T. Berners-Lee. Linked data- the story so far. *International Journal On Semantic Web and Information Systems*, 2009.