D3.1.3: TEXT PROCESSING COMPONENT

Tadej Štajner

Distribution: Public

MultilingualWeb-LT (LT-Web)
Language Technology in the Web

FP7-ICT-2011-7

Project no: 287815

Document Information

Deliverable number:	3.1.3
Deliverable title:	Text Processing Component
Dissemination level:	PU
Contractual date of delivery:	30 September 2013
Actual date of delivery:	31 October 2013
Author(s):	Tadej Štajner
Participants:	JSI
Internal Reviewer:	ENLASO, TCD
Workpackage:	WP3
Task Responsible:	Tadej Štajner
Workpackage Leader:	Clemens Weins

Revision History

Revision	**Date**	**Author**	**Organization**	**Description**
1	27/09/2013	Tadej Štajner	JSI	Draft
2	09/10/2013	Tadej Štajner	JSI	Clarity improvements, graphics
3	22/10/2013	Tadej Štajner	JSI	Incorporating comments from Y. Savourel, ENLASO, and D. Lewis, TCD
4 - final	28/10/2013	Tadej Štajner	JSI	Final version. Incorporating comments from F. Sasaki, DFKI

Contents

Executive Summary

This deliverable describes how natural language processing can be used to improve the content localization lifecycle using technologies, such as named entity disambiguation with semantic knowledge bases. We describe the requirements from the localization side to support the use case of translating named entities and the resulting design constraints. We describe a standard data representation that was defined in the ITS2.0 W3C standard that allows integration of various language tools into this process. We present a reference implementation for the selected data categories based on the Enrycher text analysis system.

1. Introduction

Translation mechanisms for named entities depend on both the source and target languages. There are specific rules to translate (or transliterate) particular proper names or concepts. Sometimes, they should not even be translated. In order to support this use case, we propose to use an automatic natural processing method to annotate the content so that it can be correctly processed.

The purpose of this work is to enable that the results of text analysis can be annotated in content. Besides translating names of entities, there are several other translation-related tasks could be improved with the help of such NLP information, for example:

- Term suggestion
- Contextualization
- Suggestion of things not to translate
- Automated transliteration of proper names

2. Requirements

In the requirements gathering phase of the standardization process we had outlined a specification and a use case for the role of text processing components in the content localization process.

In general, the purpose of automatic annotation reduces the manual cost of annotation and may increase the accuracy, consistency and comprehensiveness of such annotations. For example, the enrichment of source content with named entity annotations is one example of such an automatic process.

The goal of this work is to define and standardize an interface for text processing components in a localization workflow, supported by a reference implementation.

The initial data modelling discussions in the requirements gathering phase [1] resulted in identifying three concrete data category prototypes that were relevant for text processing tools: **sense disambiguation**, **named entity annotation**, and **annotation of text analysis**. These prototype data categories illustrate the requirements of what ITS2.0 should support.

2.1 Prototype data categories

This section outlines the prototype data categories and their functional requirements, which were later changed and consolidated into the final data categories.

SENSE DISAMBIGUATION

Definition

Annotation of a single word, pointing to its intended meaning within a semantic network. Can be used by MT systems in disambiguating difficult content.

Data model

- **meaning reference**: a pointer that points to meaning (synonym set) in a semantic network that this fragment of text represents.
- **semantic network**: a pointer (URI) that points to a resource, representing a semantic network that defines valid meanings.

The value of the semantic network attribute should identify a single language resource that describes possible meanings within that semantic network. The mechanism should allow for the validation of individual meanings against the semantic networks using common mechanisms.

The sense disambiguation, as discussed in the requirements phase, consists both of individual word senses, as well as more conceptual senses.

NAMED ENTITY ANNOTATION

Definition

Annotations of a phrase spanning one or more words, mentioning a named entity of a certain type. When describing a fragment of text that has been identified as a named entity, we would like to specify the following pieces of information in order to help downstream consumers of the data, for instance when training MT systems

Data model

- **entity reference:** a pointer (URI) that points to the entity in an ontology.
- **entity type**: a pointer (URI) to a concept, defining a particular type of the entity.

The named entity annotation proposal had a slight conceptual overlap with sense disambiguation in the requirements phase, since both are used to link textual fragments to external knowledge bases. Subsequent discussions lead to consolidation of both sense disambiguation and named entity annotation data categories into a common **text analysis** data category.

TEXT ANALYSIS RESULT ANNOTATION

This data category allows the results of text analysis to be annotated in content.

Data Model

- **annotation agent** - which tool has produced the annotation
- **confidence score** - what is the system's confidence for this annotation, on the range of [0.0, 1.0].

This prototype data category represents the requirement to specify what tool was used in a given processing step, and what the tool's estimate of the output quality is.

DOMAIN

This data category specifies the domain of the text.

Data model

- domain name

It should be able to point to multiple domains, as well as support mapping between different domain vocabularies.

All of the mentioned prototype data category requirements were consolidated and refactored into new data categories during the specification phase, namely into **Text Analysis** which covers word and entity senses, and the **Annotators reference** mechanism to represent what tool has produced a given annotation.

During the standardization process, the **domain** data category has remained relatively intact, adding only the domain mapping mechanism to accommodate different domain vocabularies.

2.2 Scope

The final requirements that were identified within the process represent a subset of what natural language processing can potentially offer to assist content processing for localisation.

However, since the purpose of the project was to implement a useful and manageable standard, we had limited our scope to the functionalities that extended existing best practices that we could test, leaving the other use cases for other related standards, such as NIF [4], which takes care of the morphosyntactic properties of individual words.

For **text analysis annotation,** we had also considered and discussed differentiating between different levels of annotations that link phrases with knowledge bases, such as distinguishing between word sense disambiguation, concept disambiguation and entity disambiguation, as well as connecting it with term disambiguation. However, another survey of requirements revealed that introducing this distinction into the data category would not support any relevant use case. Therefore, we had defined that the **its-ta-ident-ref** property can be used to represent any type of linkage between the annotated phrase and a knowledge base, making no assumptions about the type of the link.

With regard to the **domain** data category, we had identified that there was no plausible way of using a standardized domain set that could be used to validate the metadata, so the domain data category is now represented as an arbitrary string. We had also considered distinguishing between different domain axes:

topic, register, and genre. However, the consensus was that this distinction is best left to the system implementers.

2.3 Business benefits

The benefit of using text processing tools hinges on the adoption of a standardized interface that can lower the barrier to such as system. While named entity extraction has already been shown to improve translation quality [15], named entity extraction component are typically language-dependent. This often entails that they have different implementations, which increases the integration effort.

The benefit of standardizing this interface is the reduced marginal integration effort that needs to be applied for supporting a text processing components for an additional language.

3. Support in ITS2.0

The Text Analysis data category is used to annotate content with lexical or conceptual information for the purpose of contextual disambiguation.

This information can be provided by so-called text analysis software agents such as named entity recognizers, lexical concept disambiguators, etc., and is represented by either string valued or IRI references to possible resource descriptions. For example: A named entity recognizer provides the information that the string "Dublin" in a certain context denotes a town in Ireland.

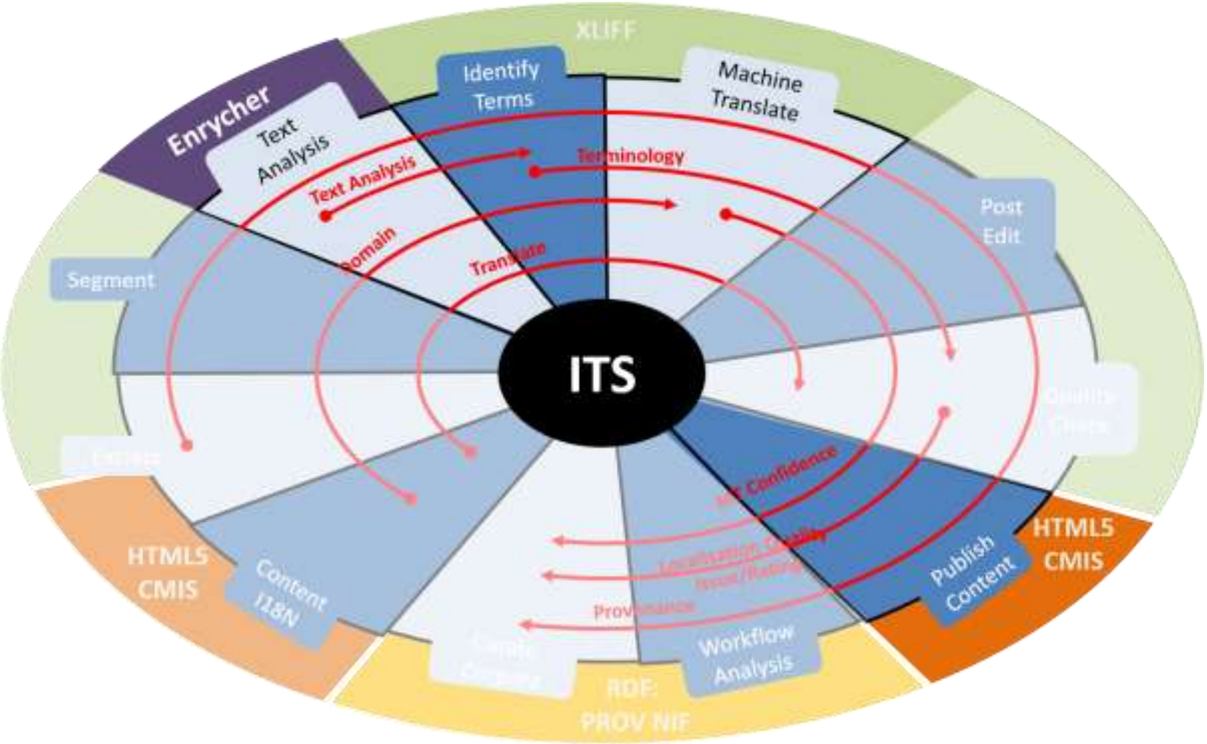


Figure 1: The role of text analysis in the ITS 2.0 ecosystem

While **text analysis** can be done by humans, this data category is targeted more at software agents. The information can be used for several purposes, including, but not limited to: Informing a human agent such as a translator that a certain fragment of textual content (so-called “text analysis target”) may follow specific processing or translation rules.

Figure 1 shows where text analysis fits in to the whole ecosystem: it feeds into terminology management, as well as machine translation pre-processing.

The ITS2.0 standard fulfilled the requirements with the following properties of the text analysis data category.

- **Text analysis confidence**: The confidence of the agent (that produced the annotation)in its own computation
- **Entity type / concept class**: The type of entity, or concept class of the text analysis target IRI
- **Entity / concept identifier**: A unique identifier for the text analysis target

These can be used in the following way in an HTML5 setting in the following fragment:

```
<!DOCTYPE html>

... <div

      its-annotators-ref="text-
analysis|http://enrycher.ijs.si/mlw/toolinfo.xml#enrycher">

      <span its-ta-ident-ref="http://dbpedia.org/resource/Dublin"
      its-ta-class-ref="http://schema.org/Place">Dublin</span> is the <span
      its-ta-ident-ref="http://purl.org/vocabularies/princeton/wn30/synset-
capital-noun-3.rdf">capital</span> of

      <span its-ta-ident-ref="http://dbpedia.org/resource/Ireland"
            its-ta-class-ref="http://schema.org/Place">Ireland</span>.
</div> ... (continued)
```

Here, the **its-annotators-ref** is used as a mechanism for annotating the provenance of certain ITS2.0 attributes, **its-ta-class-ref** is used to denote the type class of the entity behind the name 'Dublin', whereas **its-ta-ident-ref** is used to point to the DBpedia concept for the interpretation of the name Dublin. At the same time, the example also demonstrates the use of **its-ta-ident-ref** to point to an interpretation of the word 'capital'.

The **domain** data category is another example that can be supplied by automated text processing tools. While the domain can be arbitrary, a text classification model can be trained on a set of labelled documents, and then used to annotated new documents.

4. APPLICATION SCENARIOS

From the perspective of natural language, processing, there are several techniques that are used to support this use case:

4.1 Named entity extraction

Named entity extraction is a technique that identifies fragments of text that can be interpreted as named entities of certain types, such as locations, people, organizations, products, events and others. The natural language processing community also treats numeric or monetary expressions as named entities due to the fact that the same methods can be used for extraction, even though they are not 'entities' in the traditional definition of the word.

4.2 Named entity disambiguation

Using named entity extraction and disambiguation can provide links from literal terms to concrete concepts, even in ambiguous circumstances. It is inherently a difficult problem: A name can refer to many entities, an entity can have many names.

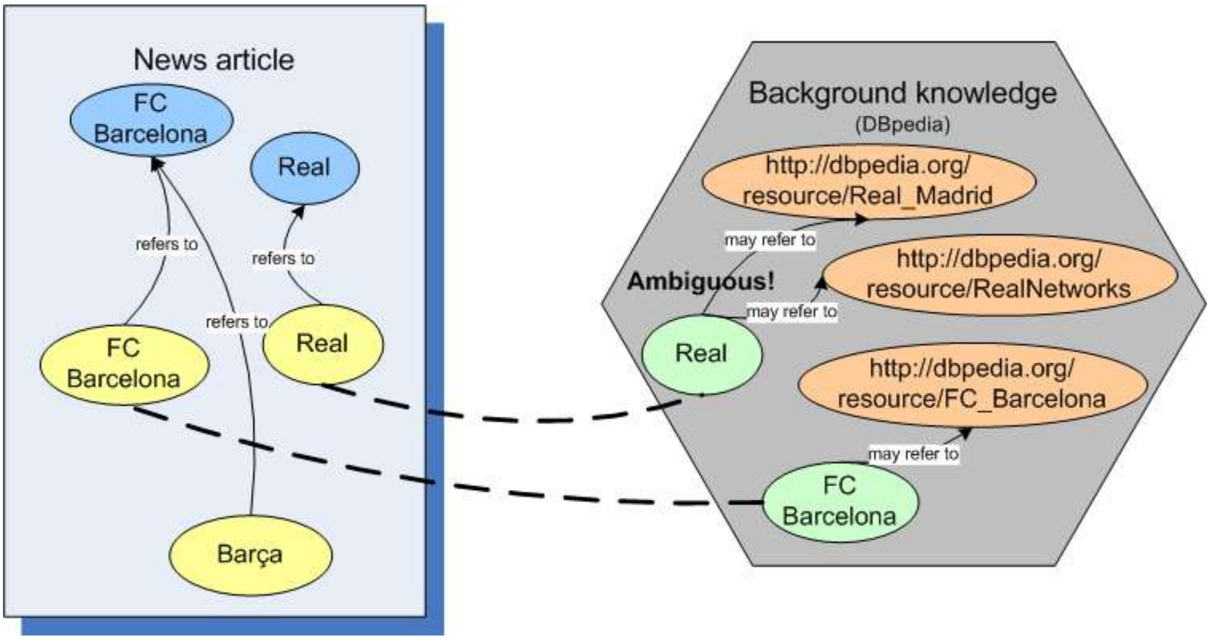


Figure 2: The problem setting of resolving the correct interpretation of ambiguous named entities.

Figure 2 illustrates the problem set-up for this situation. The system needs to be capable of deciding which interpretation of the phrase "Real" is correct. Since humans are good at this, since we have prior knowledge on the 'usual' meanings, and we can glean the meaning from the context, automated approaches try to mimic the human heuristics:

Prior knowledge using a lot of training data, and learn a probabilistic model (what is the most frequent meaning of 'London')?

Context: does the surrounding text of the mentioned name use similar vocabulary than the context of the entity? (using the word 'London' in the context of 'Canada' is likely to be interpreted as another London in Ontario).

Relational similarity: when disambiguating, take a look at the background knowledge graph and consider those connections (things that are connected in the graph tend to appear together), as illustrated in Figure 3.

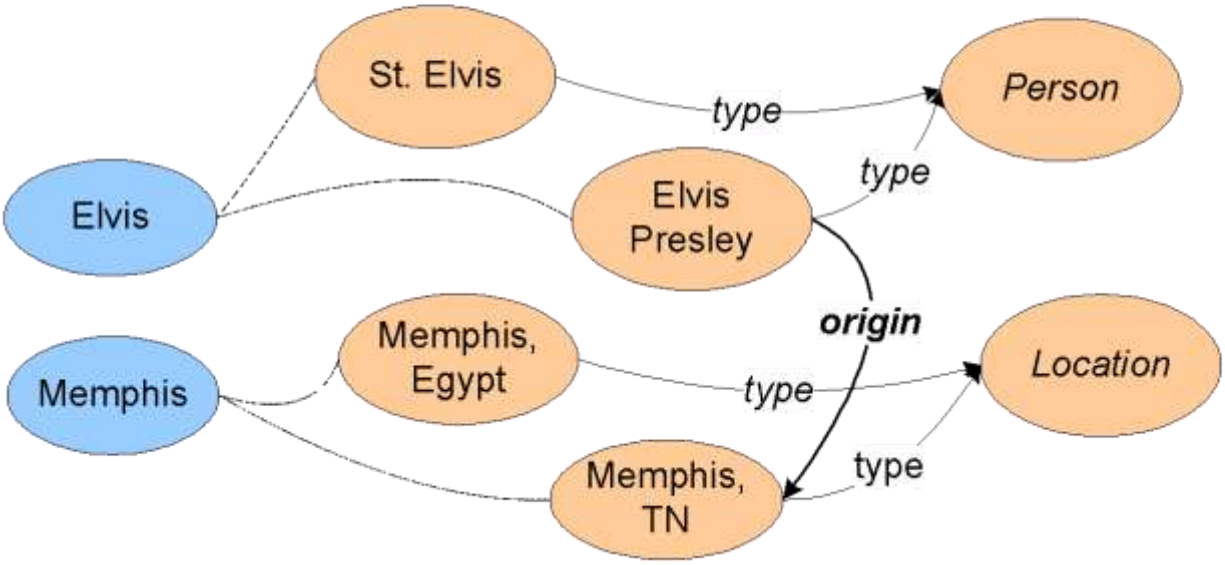


Figure 3: The set-up for relational similarity to leverage background knowledge to improve accuracy

Research [8] shows that all three signals are important in producing good disambiguation decisions, especially the relational information that is crucial in this example, where the connection that Elvis Presley was born in Memphis, TN, helps disambiguate both entities simultaneously.

4.3 Topic classification

Topic classification into a pre-defined topic ontology is a classic text analysis scenario. While there is no pre-defined ontology, we recommend implementers to train their own models using existing annotated data with available supervised text classification approaches.

5. Implementation

5.1 Reference implementation

To realize this use case, tooling is already available and will be tailored by working group participants. One main tool in this respect is the Enrycher tool. Enrycher adds metadata for semantic and contextual information. These links to concepts can be used to indicate whether a particular fragment of text represents a term, whether it is of a particular type, and alternative terms that can be used for that concept in other languages.

Enrycher [11] is a service-oriented natural language processing framework developed at JSI. It is used as a basis for the reference implementation of ITS 2.0 data categories for Text Analysis and Domain, which are available publicly [3] and are validated in the ITS 2.0 test suite [2].

Concretely, Enrycher uses DBpedia to serve as a multilingual knowledge base in order to map concepts to terms in foreign languages. Given that it also outputs the type of the term even if the exact term is not known, it can still serve as input to translation rules that apply to specific term types (personal names, locations, etc.).

Our reference implementation within Enrycher uses GATE [5] for the English named entity extraction, and an in-house implementation [7] based on Mallet [6] for the Slovene named entity extraction.

Named entity disambiguation into DBpedia is done using our implementation of an entity resolver [8]. Word sense disambiguation is done by a collective disambiguation approach [9]. Best practices demonstrate that for a general purpose implementation, DBpedia can serve as a good knowledge base for entity identifiers, especially due to its multilingual nature.

For word sense disambiguation, there are sense networks available for most languages. Our implementation for English operates on the Princeton Wordnet 3.0.

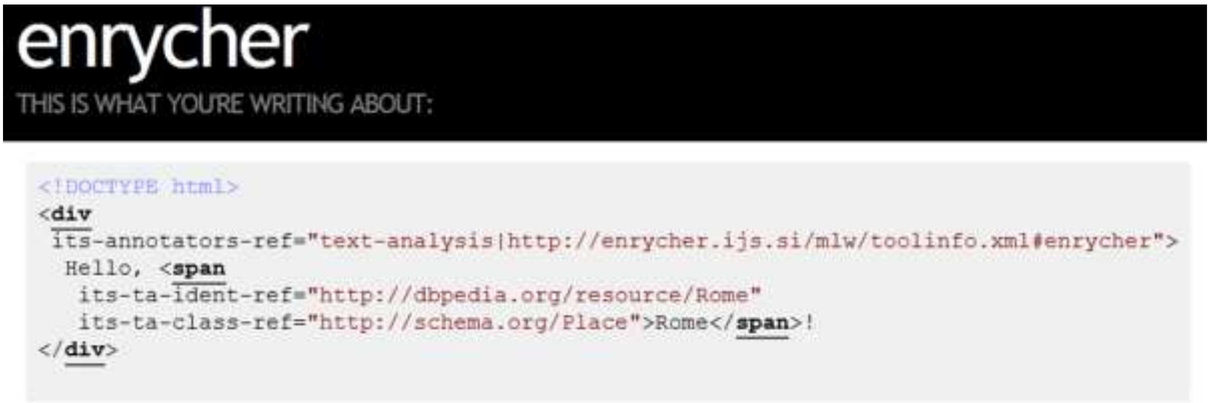
For domain classification, we use a general-domain taxonomy: the Open Directory Project (http://www.dmoz.org) ontology. Classification itself is done by a large-scale hierarchical classification approach [10]. However, most usages of domain classifications consist of internal domain taxonomies.

5.2 Use cases

Enrycher was integrated in the Okapi framework as a pre-processing component [14] for extracting text analysis annotations, which was demonstrated to the wider localization community [12], as well as the Unicode and the internationalization community [16].

It was also integrated in the Drupal Modules [13] for producing inline HTML5 annotations at authoring time.

5.3 Usage



```
<!DOCTYPE html>
<div
 its-annotators-ref="text-analysis|http://enrycher.ijs.si/mlw/toolinfo.xml#enrycher">
  Hello, <span
   its-ta-ident-ref="http://dbpedia.org/resource/Rome"
   its-ta-class-ref="http://schema.org/Place">Rome</span>!
</div>
```

Figure 4: The output from the Enrycher web service.

Since Enrycher is publically available as a web service, it can be used using standard HTTP tools. It can be accessed at http://enrycher.ijs.si/mlw/. Besides having a visual demonstration of the ITS2.0 mark-up, as shown in Figure 4, it can be also used as a REST web service using one of the following API endpoints:

- `http://enrycher.ijs.si/mlw/en/entityType.html5its2`
- `http://enrycher.ijs.si/mlw/en/entityIdent.html5its2`
- `http://enrycher.ijs.si/mlw/en/lexicalIdent.html5its2`
- `http://enrycher.ijs.si/mlw/en/entityTypeLexicalIdent.html5its2`
- `http://enrycher.ijs.si/mlw/en/entityIdentLexicalIdent.html5its2`

In this nomenclature, the **entityType** triggers the type classification, the **entityIdent** triggers the entity disambiguation, while **lexicalIdent** triggers lexical disambiguation of individual words. The input content is put in the POST request body as an HTML5 document with a MIME type of “text/html”.

For example, issuing the following terminal command for posting an HTTP request with the input HTML content in the body:

```
curl -d "<p>Welcome to London</p>"
http://enrycher.ijs.si/mlw/en/entityType.html5its2
```

Returns:

```
<p>Welcome to <span  its-ta-class-
ref="http://schema.org/Place">London</span></p>
```

as its output.

The implementation of the ITS2.0 processing code used in this demonstration is available at https://github.com/tadejs/enrycher-its20 [3].

6. Conclusions

This deliverable describes the process of standardizing the output of natural language processing components, its result and a reference implementation of the text analysis and domain data categories. We outline the steps that were necessary to standardize this effort, as well as its final form. We provide recommendations for implementation of the processing pipeline, as well as the recommendations for datasets that can be used as knowledge bases.

References

1. Multilingual Web – LT working group: ITS2.0 Requirements, http://www.w3.org/International/multilingualweb/lt/wiki/Requirements, 2012

2. Multilingual Web – LT working group: The ITS2.0 Test Suite, https://github.com/finnle/ITS-2.0-Testsuite/, 2013

3. T. Štajner: Enrycher-ITS2.0, https://github.com/tadejs/enrycher-its20, 2013

4. Integrating NLP using Linked Data. Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. 12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia, (2013)

5. H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002. PDF. BibTeX.

6. McCallum, Andrew Kachites. "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu. 2002.

7. Tadej Štajner, Tomaž Erjavec and Simon Krek. Razpoznavanje imenskih entitet v slovenskem besedilu; In Proceedings of 15th Internation Multiconference on Information Society - Jezikovne Tehnologije 2012, Ljubljana, Slovenia

8. Tadej Štajner and Dunja Mladenić. 2009. Entity Resolution in Texts Using Statistical Learning and Ontologies. In PROCEEDINGS OF THE 4TH ASIAN CONFERENCE ON THE SEMANTIC WEB (ASWC '09), Asunción Gómez-Pérez, Yong Yu, and Ying Ding (Eds.). Springer-Verlag, Berlin, Heidelberg, 91-104.

9. Rusu, D., Stajner, T., Dali, L., Fortuna, B. and Mladenic, D. 2010. Enriching Text with RDF/OWL Encoded Senses. Demo. 9th International Semantic Web Conference (ISWC 2010). Shanghai, China.

10. Grobelnik, Marko, and Dunja Mladenić. "Simple classification into large topic ontology of web documents." *Journal of Computing and Information Technology* 13.4 (2004): 279-285.

11. Štajner, T., Rusu, D., Dali, L., Fortuna, B., Mladenić, D., & Grobelnik, M. (2010). A service oriented framework for natural language text enrichment. *Informatica (Ljublj.)*, *34*(3), 307-313.

12. Savourel, Y, Štajner, T.: Creating Translation Context with Disambiguation, Localization World, 2013, London, UK.

13. Fritsche, K, Walter, S. LT-Web Deliverable D3.1.1: Drupal Modules, 2013

14. Savourel, Y. LT-Web Deliverable D3.1.4: Okapi Components for XLIFF, 2013

15. Babych, Bogdan, and Anthony Hartley. "Improving machine translation quality with automatic named entity recognition." Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT. Association for Computational Linguistics, 2003.

16. Sasaki, F, Lieske, C: ITS 2.0: Facilitating Automated Creation and Processing of Multilingual Web Content, Internationalization and Unicode Conference 37, Santa Clara, 2013