Information Extraction in Semantic Wikis

Pavel Smrž and Marek Schmidt

Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 66 Brno, Czech Republic
E-mail: {smrz,ischmidt}@fit.vutbr.cz

Abstract. This paper deals with information extraction technologies supporting semantic annotation and logical organization of textual content in semantic wikis. We describe our work in the context of the KiWi project which aims at developing a new knowledge management system motivated by the wiki way of collaborative content creation that is enhanced by the semantic web technology. The specific characteristics of semantic wikis as advanced community knowledge-sharing platforms are discussed from the perspective of the functionality providing automatic suggestions of semantic tags. We focus on the innovative aspects of the implemented methods. The interfaces of the user-interaction tools as well as the back-end web services are also tackled. We conclude that though there are many challenges related to the integration of information extraction into semantic wikis, this fusion brings valuable results.

1 Introduction

A frequently mentioned shortcoming of wikis used in the context of knowledge management is the inconsistency of information that often appears when wikis are put to everyday use of more than a few knowledge workers. Semantic wikis, combining the easy-to-participate nature of wikis with semantic annotations, have a strong potential to help in this situation and to become the ultimate collaborative knowledge management system. However, adding metadata means additional work and requires user's attention and thinking. Since it is often difficult to give users immediate satisfaction in reward for this tedious work, annotations in internal wikis tend to be rather scarce. This situation has a negative impact on comprehension of the advantages of semantic wikis and discourages their extensive deployment.

Information extraction (IE) can be seen as a means of reducing user's annotation workload. It refers to the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources [19]. The state-of-the-art IE technology can produce metadata from content, provide users with useful suggestions on potential annotations and ask questions relevant for the current context. The ultimate goal of IE in semantic wikis is to maximize the benefits of the rich annotation and, at the same time, minimize the necessity of manual tagging.

New application domains raise various challenges for large-scale deployments of IE models. Despite more than two decades of intensive research, the accuracy of the systems is still unsatisfactory for many tasks. Moreover, the results strongly depend on the domain of applications and the solutions are not easy to be ported to other domains. Language dependency is also an issue as the level of analysis required by some methods is available only for a few languages. Another difficulty, particularly significant for the use of IE technology in semantic wikis, lies in the limited character of examples that could be used to train extraction models. Indeed, the real use of semantic technologies calls for specialized annotations of complex relations rather than simple and frequent entities such as places, dates etc. Users are not willing to look for more than one or two other occurrences of a particular relation that should be automatically tagged.

The issues related to standard IE solutions also determine the work described in this paper. As almost all realistic IE-integration scenarios involve system suggestions and user interaction, the IE components that have been designed and are successively developed can be taken as a kind of semantic wiki recommendation system. We pay a special attention to the "cold-start problem" which appears in the beginning of the technology deployment when there are no data to provide high quality suggestions. Taking the semantic wiki as an open linking data platform (rather than a tool to enrich data with semantics for internal purposes only) helps in this respect as one can immediately take advantage of external data sources. We also deal with the incremental character of IE tasks running on gradually growing wiki pages. The implemented interfaces of the IE services facilitate the process of continuous updating of the annotations. They also help to interlink external resources that are modified independently of the controlled updates in the internal wiki.

The following sections tackle the mentioned challenges and show how IE can be used in real semantic wiki systems. As a use case, the next section briefly introduces the IE techniques and tasks applied in the KiWi project. Section 3 discusses specific features of the IE techniques required by the semantic wikis and the innovative aspects of the KiWi solutions. We conclude with future directions of our work.

2 IE Techniques and Tasks in the KiWi project

2.1 Conceptual Model

The main objective of the KiWi (Knowledge in a Wiki[1]) project is to facilitate *knowledge sharing* and *collaboration* among the users of the system to manage knowledge in a more efficient way [20]. Together with personalization, reasoning and reason maintenance, IE belongs to the key enabling technologies in KiWi.

There are two main use cases in this project. The first one is provided by Sun Microsystems, Prague, and is focused on knowledge management in software development, particularly in the NetBeans project. The second one addresses

[1] `http://www.kiwi-project.eu`

vital issues in process management in Logica. The examples given in this paper are taken from those use cases.

KiWi allows users to add meta-data to individual pages or their parts in the form of free or semantic tags. The role of IE is to support users in creating the semantic meta-data and making the knowledge explicit so that it can be further queried and reasoned in a semantic way. The conceptual model for IE in KiWi consists of three major components:

- content items;
- text fragments;
- annotations of content items.

Content item refers to any entity that can be identified. Text fragment is an arbitrary continuous piece of a content item. Text fragments are content items themselves. It enables adding metadata to individual text fragments. In a simple case of commenting a piece of information on a wiki page, the metadata can be of type "comment" and can contain the text of the comment. Tagging text fragments provides a bridge between structured and unstructured information. The fragments can be taken as generalizations of links representing any kind of related resources. In that sense, the fragments are independent of the formatting structure of the wiki page.

Figure 1 demonstrates the way in which information extracted from three different content items (wiki pages) is put together. All the pages mention the same resource – an issue identified by its number #1223 (corresponding to Sun's bug-reporting site IssueZilla). In the "Description of Issue #2536", one can read that the issue is actually a consequence of Issue #1223. The page on "Release Engineering" page says that issue #1223 is critical for the current release of the final product. Finally, "Meeting minutes" assign the task to fix the bug to Joe. The information extraction component is able to extract the mentioned pieces of information and save them (as a set of RDF triples) for further processing.

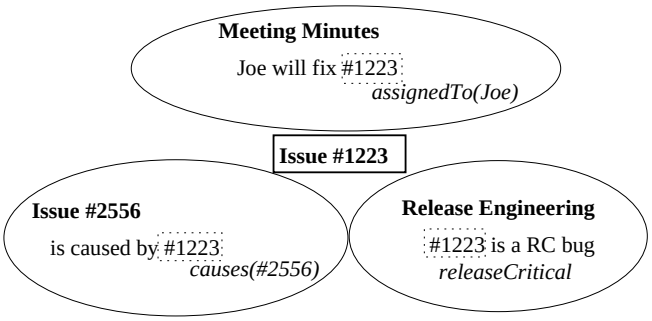

Fig. 1. Information about one entity may come from different pages

Note that the described concept of tagging text fragments that represent a resource (rather than to simply join the ascertained statement to the particu-

lar resource) enables identifying sources of extracted information, synchronizing text and tags, keeping track of the consequences of the changes and pointing out inconsistencies. What is even more important for the semi-automatic IE processes, it also makes the manual corrections easier and allows users to improve the IE accuracy by providing explicit feed-back to the system's decisions.

KiWi is designed as a modular system, in which modules provide additional functionality to the core system via *widgets* which a user may add to her custom layout. The main interaction between the IE system and the user is realized by the *annotation widget*. Figure 2 demonstrates the use of the widget for IE from meeting minutes. It shows information extracted from the text fragment appearing on the wiki page. The widget also offers actions enabled by the semantics of extracted information (such as inserting the event to the calendar, showing the map of the place, etc.). The annotation editor that has been also developed allows users to manually annotate fragments directly in the KiWi editor and to inspect associated semantic information in the annotation widget.

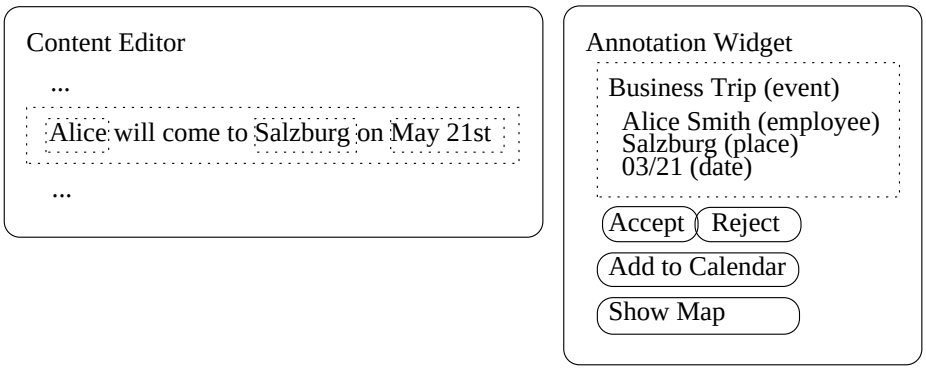

Fig. 2. An example of metadata and action suggestions provided by the KiWi annotation component

KiWi aims at an application of the state-of-the-art IE methods in the semantic wikis [8]. Various techniques and methods are employed to fulfil the key tasks identified in the project.

IE from wiki pages deals mostly with free text. In general, it can therefore benefit from a thorough language analysis of the input. In addition to tokenization and sentence splitting, the natural language processing may involve stop-list filtering, stemming or lemmatization, POS tagging, chunking and shallow or deep parsing. Many of these pre-processing steps are computationally expensive. Almost all of them are strongly language-dependent. Moreover, there is danger of cascading of errors from pre-processing in the IE system. That is why we follow the current trend and apply selective pre-processing in KiWi. The methods described below take into account the context in which they work and employ only those pre-processing processes that can bring significant value to the IE itself.

2.2 Implemented IE Techniques

The IE solutions implemented in KiWi rely on several state-of-the-art IE techniques. Before discussing the function of particular IE processes in KiWi, let us therefore mention the crucial techniques employed. The following technologies play a key role in the system:

- automatic term recognition combining domain-specific and general knowledge;
- computation of word relatedness to define similarity measures;
- text classification and sense disambiguation based on advanced machine-learning methods;
- dimensionality reduction for text feature vectors.

Any realistic approach to automatic term recognition (ATR) from wiki pages cannot ignore the fact that the source texts are usually rather short. Unfortunately, most of available ATR methods rely too much on high frequency counts of term occurrences and, therefore, cannot be utilized in the intended field.

To cope with the problem, we adopt a new ATR method proved to give the best r results suggestesults in our previous experiments (see [9]). It flexibly combines the frequency-based measure (a variant of the TF.IDF score) and the comparisons with a background corpus. The current implementation works with a general background data (such as American GigaWord [5] or Google TeraCorpus [1] for English) only. Our future work will focus on an automatic identification of supplementary in-domain texts that would be useful for the "focused background subtraction".

Various approaches to characterize the semantic distance between terms have been also explored in our research. For general terms, we make use of the wordnet-based similarity measures [16] that take into account the hierarchical structure of the resource. The same technique is employed when the closeness of concepts in a domain-specific thesaurus or ontology is to be computed (e.g., on Sun's Swordfish ontology [3]).

An implemented alternative method which does not require manually created resources (such as wordnet-like lexical databases or domain ontologies) determines the semantic similarity of terms by the relative frequency of their appearance in similar contexts. Of course, there are many ways to assess the similarity of contexts. The results of our preliminary experiments suggest that the best performer for the general case is the method taking into account the (dependency) syntactical structure of the contexts [7, 10] (terms are semantically close if they often appear in the same positions, e.g., as subjects of the same verb, modifiers of the same noun, etc.).

Many IE tasks can be formulated as classification problems. This finding is behind the immense popularity of machine learning techniques in the IE field today. In the rare case when there are enough data for training, KiWi follows this trend. Complex features computed on the dependency structures from the source text are gathered first.

The particular set of features applied depends on the task and the language in hand. For English named entity recognition, a gazetteer, word contexts, lexical and part of speech tags are used. For classification of the role an entity plays on a page (which can be interpreted as a semantic role labeling problem [13]), additional features provided by a dependency parser are employed. The classification is performed by CRF (Conditional Random Fields) and SVM (Support Vector Machine) models with tree kernels constructed from syntax trees of the sentences [21]. Depending on the context, the process can identify "soft categories" sorted by the descending probability of correctness. The resulting N-best options are presented to the user who chooses the correct one.

As opposed to the discussed situation, a typical classification task in the context of semantic wikis can be characterized by the limited character of the input text and the lack of data to train the classifier. The advanced methods that can deal with the latter issue are discussed in the next section. Let us therefore just note, that to overcome the former one (inherent to the wiki world), KiWi harnesses the other mentioned techniques and personal/organizational contexts to characterize the "ground" of the material provided by the user and to increase accuracy of the classification.

As exemplified by the Logica use-case in KiWi, the semantic wikis in the professional setting often need to integrate large sets of business documents (product specifications, customer requirements, etc.). Having such a document in hand, the user can ask the system to find similar documents in the given collection. As the terminology and the style of the documents can differ significantly, the straightforward computing of the similarity as a function of term co-occurrences is often insufficient. Standard approaches to overcome this (such as PLSA – Probabilistic Latent Semantic Analysis or LDA – Latent Dirichlet Allocation) transform the term vectors representing the documents to point out their semantic closeness.

Unfortunately, the computation of such transformations is prohibitively expensive. KiWi draws on the random indexing technique [6] that is several orders of magnitude faster than the mentioned approaches. As KiWi documents are indexed by means of Apache Lucene search library – we take advantage of Semantic Vectors [22] – a fast implementation of the random indexing concept based on the Lucene indices. This setting provides very efficient mechanism to evaluate similarity queries in KiWi.

2.3 IE Tasks in KiWi

The above-mentioned IE techniques find their application in various tasks and various contexts in the KiWi system. From a general point of view, the whole IE functionality can be seen as tag suggestion or automatic annotation (if the similarity is interpreted as a special kind of tagging). On the other hand, the user perspective distinguishes different kinds of tags for different purposes. The following tasks form the core of the KiWi IE module in the latter sense:

- suggestion of new free-text tags and thesaurus/ontology extensions;

– entity recognition and semi-automatic annotation of content items;
– relation extraction and structured tag suggestion;
– similarity search adapted according to the user's feedback.

Figure 3 (based on the KiWi use case defined by Logica) demonstrates the interplay of these tasks. It shows a situation when a project manager comes to the task to produce a project risk analysis report based on her notes from a preparatory meeting (as displayed in the KiWi annotation editor on the left side of the picture). Risk-related information needs to be formalized, the potential impact should be identified and the resolution strategies explicitly stated. Based on the user-specific setting, the IE component automatically identifies entities such as company products, development technologies, names of employees, dates, places, etc. and classifies the page as a (seed of) risk analysis report – a known type of document with an associated semantic form. The identified type narrows down the similarity search which focuses on those risk analysis reports that mention semantically related risks (it is realized as a simple word-based relatedness function on the "identified risks" sections in the current implementation).

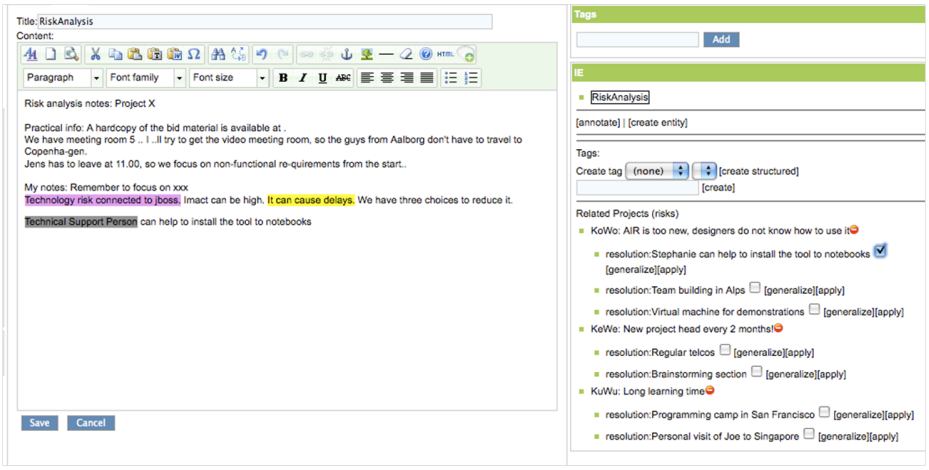

Fig. 3. KiWi annotation component classifying entities and relations and suggesting tags and projects related to a given page according to the identified risks

The annotation component also suggests terms found in the text as additional tags. The possibility to propose free-text tags is not particularly useful in the semantically-rich case discussed but it can be essential for "lightweight" semantic wiki environments. A more practical function in the actual context refers to semi-automatic extending the conceptual domain model. The most frequent form of this process regards thesaurus or ontology population by instances referred to in the analysed text. For example, finding new term "JBoss Seam" in the position where a development tool name is expected, the system can suggest adding the

term as an instance of class "development tools". The problem domain ontology can also be extended in higher levels, e.g., "video meeting room" can be suggested as a subclass of "meeting room".

Entity recognition employs the FSA (finite-state automaton) technology and implements a straightforward gazetteer strategy when it is tightly coupled with the annotation editor to identify types and instances of entities mentioned in the text and to suggest annotations linking the specific reference to the knowledge base. A limited set of rules is applied to identify compound expressions such as names, absolute temporal terms, monetary and other numeric expressions, etc. Apart from that, the functionality is completely based on lists of entities that should be identified in the texts. The lists are populated by terms referring to concepts in general ontologies (e.g., UMBEL[2] or GeoNames[3]) as well as domain-specific resources (such as Sun's Swordfish ontology or a list of team members and their roles). For the fast on-line processing, these extensive lists are compiled to a large FSA which is then used to identify matches in the text and to provide the type of the suggested tag.

Similarity search takes advantage of pre-computed matrices of term relatedness. This is crucial especially for comparing short text fragments such as the "identified risk" sections discussed above. Particular matrices correspond to various measures of the semantic distance between terms. Except for the batch document clustering, the similarity search is always intended for the on-line mode. The pre-computation of the term similarities in the form of the matrices helps to speed up the search significantly.

For the fast search on short text fragments (less than a paragraph), KiWi computes the information gain of the terms appearing in the text. The lines corresponding to the most informative terms are taken from the term-closeness matrices. This provides a simple query-expansion mechanism. The query combining the terms from the actual fragment and the semantically close terms (weighted by the informativeness and similarity, respectively) is evaluated on the all content items of the same type and the best matches are retrieved.

The whole wiki pages and full documents are indexed by the Lucene library. To perform similarity search on this kind of documents, SemanticVectors [22] are employed. It is often the case that the retrieved documents do not come up to user's expectations. The most informative terms can prove to be unimportant from the user's perspective. That is why it is very important to let KiWi users know why the particular documents are considered similar to that one in question and what terms played the key role in the system's decision. KiWi lists those terms for the entire set of the similar documents and for each individual document as well. The user can mark some of the terms as unimportant for the current context and the system re-computes the similarity with the new restricted set of terms.

The concept of tags in KiWi is rather general. It comprises the standard label-like tags, but also structured ones that encode relations of the concept

[2] `http://www.umbel.org`
[3] `http://www.geonames.org`

represented by the given term to other concepts. The corresponding IE task of relation extraction extracts facts from relations between entities in a wiki page (e.g., from statements like *Alice will travel to Salzburg on May 21st*). The relation can also be identified between an entity and the wiki page itself, since every page in the KiWi represents some entity.

The implementation of the relation extraction algorithm is similar to that of entity recognition. It employs advanced machine learning models (CRF mentioned above) and incorporates additional information provided by the user to improve the performance. For example, the user can specify features relevant for semi-structured documents as an XPath expression (e.g., to inform the automatic extraction method that the cost is always in the second column of a table). Unfortunately, the process is prone to the errors in the language analysis layer so that the results strongly depend on the quality of the language-dependent pre-processing phase.

Semantic wikis with annotations support evolution of knowledge from free-form to structured formalized knowledge. The role of IE is to support the user in the process of creating semantic annotations. If the structure of knowledge is well understood, the annotations can take a unified form of tags anchored in a domain ontology. However, the "wiki way" of knowledge structure that is only emerging in the collaborative work process calls for sharing of free-text tags as well. KiWi supports this by means of new tag suggestions based on the ATR (see above) from a particular document or a wiki page. Users can choose which extracted terms are appropriate to tag the resource and what their relations to other tags are. For ATR on short wiki pages, KiWi engages heuristics based on simple grammar patterns (such as "an adjective followed by a noun") to propose the candidate terms.

In addition to free-text tagging, ATR makes it also possible to suggest extensions to a domain ontology or thesaurus. KiWi checks whether the extracted terms correspond to existing concepts and if not, it proposes additions. If there are enough data for classification training, it can also find the most probable class to link the new concept to.

3 Innovative Aspects of IE in KiWi

As mentioned above, there are many challenges and open questions related to the use of IE in semantic wikis. The state-of-the-art IE systems [2, 12, 17] often make assumptions about the type of data, its size and availability, and the user interaction mode that are not acceptable in the given context. KiWi explores solutions that are able to cope with the problems and work the "wiki way" (provide sophisticated functionality but easy to understand and easy to use).

Machine learning plays a central role in the current IE paradigm [14]. From a conceptual point of view, statistical IE systems distinguish two phases: the *training phase* and the *deployment phase*. In the training phase the system acquires a model that covers a given set of annotation examples. In the deployment phase, the system identifies and classifies relevant semantic information in new

texts, i.e., texts that were not included in the training set. The predominant approach expects a large text corpus with annotated information to be extracted, and then uses a learning procedure to extract some characteristics from the annotated texts [23]. Unfortunately, an annotated training data set is available for a very limited number of cases. And it is unrealistic to expect that KiWi users will provide this kind of data to make the system "semantics-aware". This is especially true for the case of many application-specific relations in the explored domains.

To overcome the problem of training data scarcity, IE in KiWi explores a combination of the standard supervised algorithms with the methods that are able to learn from untagged texts. We take advantage of the concept of *bootstrapping*, which refers to the technique that starts from a small initial effort and gradually grows into something larger and more significant [14]. One of the currently employed methods relying on this principle is the *expansion*. An initial extraction model (learned from few examples) is applied to the unannotated text (wiki pages, linked documents or external resources) first. Newly discovered entities or relations that are considered sufficiently similar to other members of the training set are added and the process is iterated.

Another approach we apply is *active learning*. In active learning, the system itself decides what the best candidates for annotation are in order to maximize the speed of the learning process. A user is then asked to annotate these instances only. The idea of active learning perfectly fits the wiki philosophy that every user can annotate every page for which she has sufficient rights. All changes are naturally reported and there is no problem to come back to a previous version in case somebody made inappropriate annotations.

The combination of both methods lets the system exploit the knowledge as much as possible, but still allows users to have full control of the annotation process.

There is not much to do about the dependency of the IE methods on the result of the pre-processing phase. The trade-off between the quality of the language analysis and the general availability of the corresponding tools makes it impossible to provide the same grade of extraction in all languages. KiWi tries to mitigate the "curse of language-dependency" by means of using general resources that are available across languages. For example, our experiments with instances of Wikipedia in several languages used for the expansion proved that this functionality does not need to be limited to a particular language.

In addition to the lack of annotated data for training classifiers, there is also a specific problem of the unusual nature of some IE tasks in semantic wikis. The resources that are to be semantically annotated vary exhibit high diversity. The length ranges from a few words to entire pages and full documents that are uploaded to the system. Especially the lower side on this scale (very short texts) trouble the commonly used IE techniques – they often need more material to find similar contexts, to disambiguate a term, to classify a relation, etc.

One of the techniques that partially deals with the problem of short texts benefits from the PLSA and random projection algorithms discussed above. It

projects the dimensions given by the original set of terms to the space defined by a referential collection of resources. In the case of KiWi, pages from Wikipedia are taken as the dimensions. Thus, it is possible to present the results to the user in an intuitive form – pointing out the articles with the most significant contribution.

The concept of KiWi as the open linking data platform has been already mentioned. The IE technology tries to re-use as much as possible from existing semantic web knowledge resources. Dbpedia and Wikipedia find their place in the training of classifiers and sense disambiguators, the taxonomy based on WordNet and OpenCyc help to define the similarity measures etc. The external data sources are also linked to the user-interaction mode in KiWi. For example, a user defines new semantic tag "programming language" as `http://umbel.org/umbel/semset/en/wikipedia/Programming\_language`. The system fetches all relevant articles from Freebase and trains an initial classifier. The user can start to tag with it immediately and to provide feedback to improve the model.

4 Conclusions and Future Directions

Let us summarize the major point of the work reported in this paper. The application of IE methods on the specific set of problems (texts of varying size and character, complex relations, etc.) with this kind of user interface (semi-automatic, generic, ontology based) is novel. In addition to other results, KiWi brings valuable insights into the practical applicability of the best IE techniques in real conditions.

KiWi promises an advanced knowledge management system with state-of-the-art personalization, reasoning and information extraction features. As the project is still in its early phase (the first year finished in February 2009), only the first open source pre-release of the core system is available for real use. The IE components applicable in the context of the mentioned use-cases have been developed in parallel and are available in the experimental mode.

The accuracy of the IE methods significantly depends on the domain, task and data that can be used. For example, the reported figures for entity recognition range from 60 % to 90 % and generally correspond to the type of entities to be extracted [15]. The precision of the relation extraction task demonstrates even more significant variability (e.g., [18] reports results ranging from 7 % to 90 % on various relations from Wikipedia). It has been shown that the IE process can be useful even if the performance is imperfect [4]. However, to the best of our knowledge, no studies assessed the actual added value of the IE solutions for the highly interactive scenarios which is typical for the semantic wikis. This forms one of the key directions of our future work.

Another challenge we have to face in the next stage comes from the fact that the types of entities and relations to extract are not specified in advance. Users can apply the services to extract information from arbitrary complex texts. They can also specify an ontology and ask the system to identify any given relation. While, e.g., the use of ontologies to drive the IE process has been

already explored [11], it is not yet clear whether the performance of the general IE system, capable of extracting any type of entity or relation only by learning from the user annotations, will be acceptable for the end-users.

Acknowledgement

The work presented in this paper has been supported by European Commission, under the ICT program, contract No. 211932 and under the IST program, contract No. 27490.

References

1. Brants, T., and Franz, A. Web 1T 5-gram Version 1, 2006. Linguistic Data Consortium, Philadelphia.
2. Ciravegna, F., Chapman, S., Dingli, A., and Wilks, Y. Learning to harvest information for the semantic web. In *Proceedings of the 1st European Semantic Web Symposium, Heraklion, Greece* (2004).
3. Cone, S., and MacDougall, K. Case Study: The swoRDFish Metadata Initiative: Better, Faster, Smarter Web Content, 2007. `http://www.w3.org/2001/sw/sweo/public/UseCases/Sun/`.
4. Feldman, R., and Sanger, J. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* Cambridge University Press, December 2006.
5. Graff, D. English Giga-word, 2003. Linguistic Data Consortium, Philadeplhia.
6. Kanerva, P., Kristoferson, J., and Holst, A. Random Indexing of Text Samples for Latent Semantic Analysis. In *22nd Annual Conference of the Cognitive Science Society* (2000), Erlbaum.
7. Kilgarriff, A., Rychly, P., Smrz, P., and Tugwell, D. The sketch engine. In *Practical Lexicography: A Reader*, T. Fontenelle, Ed. Oxford University Press, USA, 2008.
8. Knoth, P., Schmidt, M., and Smrž, P. KiWi deliverable d2.5: Information Extraction – State of the Art, 2008. `http://wiki.kiwi-project.eu/multimedia/kiwi-pub:KiWi_D2.5_final.pdf`.
9. Knoth, P., Schmidt, M., Smrž, P., and Zdráhal, Z. Towards a Framework for Automatic Term Recognition. In *Proceedings of Znalosti (Knowledge) 2009* (2009).
10. Lin, D. Automatic Retrieval and Clustering of Similar Words. In *COLING-ACL* (1998), pp. 768–774.
11. Maedche, E., Neumann, G., and Staab, S. Bootstrapping an ontology-based information extraction system. In *Studies in Fuzziness and Soft Computing, Intelligent Exploration of the Web* (2002), Springer.
12. McDowell, L. K., and Cafarella, M. Ontology-driven information extraction with ontosyphon. In *Proceedings of the International Semantic Web Conference* (2006), pp. 428–444.
13. Mitsumori, T., Murata, M., Fukuda, Y., Doi, K., and Doi, H. Semantic role labeling using support vector machines. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)* (Ann Arbor, U.S.A., 2005), Association for Computational Linguistics, pp. 197–200. `http://www.lsi.upc.es/~srlconll/st05/papers/mitsumori.pdf`.

14. Moens, M.-F. *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer, 2006.
15. Nadeau, D., and Sekine, S. A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes* (2007).
16. Pederson, T., Patwardhan, S., and Michelizzi, J. WordNet::Similarity - Measuring the Relatedness of Concepts, 2004. `http://www.d.umn.edu/~tpederse/similarity.html`.
17. Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., and Kirilov, A. KIM - A semantic platform for information extraction and retrieval. *Natural Language Engineering 10*, 3-4 (2004), 375–392.
18. Ruiz-Casado, M., Alfonseca, E., and Castells, P. From wikipedia to semantic relationships: a semi-automated annotation approach. In *Proceedings of SemWiki06* (2006).
19. Sarawagi, S. Information Extraction. *Foundations and Trends in Databases 1*, 3 (2008), 261–377.
20. Schaffert, S. The KiWi Vision: Collaborative Knowledge Management, powered by the Semantic Web, 2008. `http://www.kiwi-project.eu/index.php/kiwi-vision`.
21. Settles, B. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)* (Geneva, Switzerland, 2004). `http://pages.cs.wisc.edu/~bsettles/pub/bsettles-nlpba04.pdf`.
22. Widdows, D., and Ferraro, K. Semantic Vectors: A scalable Open Source package and online technology management application. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)* (Marrakech, Morocco, 2008), ELRA, Ed. `http://code.google.com/p/semanticvectors`.
23. Yangarber, R., and Grishman, R. Machine learning of extraction patterns from unannotated corpora: Position statement. In *Proceedings of Workshop on Machine Learning for Information Extraction* (2001), pp. 76–83.