Grant Agreement Number: 257528

KHRESMOI

www.khresmoi.eu

Report on and prototype of multilingual information extraction

Deliverable number	*D4.1.2*
Dissemination level	*Public*
Delivery date	*1 February 2012*
Status	*Final version*
Author(s)	*Jan Dědek, Jan Hajič, Pavel Pecina*

SEVENTH FRAMEWORK PROGRAMME

This project is supported by the European Commission under the Information and Communication Technologies (ICT) Theme of the 7th Framework Programme for Research and Technological Development.

Executive summary

This deliverable concludes the effort made within the multilingual work-package (WP4) on the task of information extraction, especially in the context of deep linguistic analysis. A previous deliverable, D4.1.1, gives an overview of multilingual information extraction (IE) technologies. We provide here additional discussion and investigation of these technologies and the possible application of deep language models, originally built for tree-based machine translation, to information extraction.

The deliverable describes an extension and evaluation of the Czech multilingual information extraction prototype. Gold standard evaluation collection was constructed from the Bibliographia Medica Čechoslovaca database. A collection of nearly 25,000 Czech documents was downloaded from the internet. About 14,000 of the documents were filtered out to ensure reasonable quality. The documents were processed by linguistic tools (CUNI's sentence splitter, tokenizer and morphological analyzer) and by the Czech prototype in four different settings. Compound term analysis was performed in two of the settings. The system performance was evaluated separately for each setting and the results were compared, discussed and a final recommendation was made not to use deep language models within the project because it does not seem feasible for that kind of requirements that the project imposes. The improvement would not be significant and degradation of time efficiency expected.

Table of Contents

List of Abbreviations

ACE: Automatic Content Extraction

BDM: balanced distance metric

BMČ: Bibliographia Medica Čechoslovaca

CUNI: Charles University in Prague

FGD: Functional Generative Description (of Language)

HON: Health On the Net

IE: information extraction

ILP: Inductive Logic Programming

LDR: language dependency representation

LREC: International Conference on Language Resources and Evaluation

MARVIN: Multi-Agent Retrieval Vagabond on the Internet

MARVIN/WRAPIN: MARVIN with additional functionalities developed within the WRAPIN project

MeSH : Medical Subject Headings

MIMIR: Multi-paradigm Information Management Index and Repository

ML: machine learning

PDT: Prague Dependency Treebank

SVM: support vector machines

USFD: The University of Sheffield

ÚFAL: Institute of Formal and Applied Linguistics, CUNI

WRAPIN: Worldwide online Reliable Advice to Patients and Individuals

1 Introduction

This deliverable concludes the effort made within the multilingual work-package (WP4) on the task of information extraction, especially in the context of deep linguistic analysis. This is the second deliverable of that kind. In the first one (D4.1.1) there was a rich overview of related technologies available from different partners, namely:

- the English information extraction system developed within the WP1 work-package,
- MARVIN/WRAPIN, HON's medical multilingual service,
- Czech IE module based on GATE, CUNI's linguistic tools and MeSH taxonomy and
- MIMIR: Multi-paradigm Information Management Index and Repository developed at USFD.

These technologies were used to create a corpus of English and Czech biomedical documents, analyse and index it.

The present deliverable provides additional investigation of the possible application of deep language models, originally built for tree-based machine translation, to information extraction within the project; the Czech information extraction module was further developed and evaluated, which makes possible more qualified decision about the usage of these models.

The deep language models, originally built for tree-based machine translation at CUNI provide a complex and highly structured representation of natural language text. They provide a formal framework for representation of language meaning, and information extraction techniques can use such framework in different ways, depending on the kind of the actual extraction task. Section 2 briefly describes several information extraction tasks that are further discussed in Section 3. General overview of IE approaches using deep language parsing is provided there as well as alternative deep language representations. Discussion about the benefit of deep language parsing for different extraction task is provided by Section 3.3 and Section 3.4 discusses it from the perspective of the project needs. The dataset of Czech biomedical documents with manually assigned annotations is described in Section 4. The extended Czech information extraction prototype is described in Section 5. Evaluation of the prototype on the dataset is provided in Section 6 and Section 7 concludes the deliverable.

2 Information Extraction: Tasks and Terminology

The term *information extraction* is often used in different contexts with slightly different meanings. It is necessary to specify the term more precisely, because these disciplines cannot be always solved by the same technology and some approaches are only suitable for some disciplines.

Three terms (corresponding to different information extraction disciplines) that need to be distinguished for the purpose of this document (Section 3.3 mainly) are:

- (Named) Entity Recognition,
- Instance Resolution and
- Relation and Event Extraction.

Entity Recognition or Named Entity Recognition corresponds to the extraction task of identification of significant entities (people, organizations, locations, chemicals, genes, etc.) that are present in text.

Instance Resolution aims at linking a particular entity to its unique representative or identifier; disambiguation is the main challenge here. There is a big technological difference between general entity recognition and entity recognition that is necessarily (forced by task) coupled with instance resolution. The difference between the two tasks can be illustrated on an example where "George Bush" appears in text. General entity recognition system just marks the string and assigns a corresponding label (e.g. person, politician or president – depending on the granularity of the system) to it. For an instance resolution extraction system there is no need to assign labels to entities because such information (and usually much more than that) is saved along with the unique representatives in the system database (or ontology). However instance resolution extraction system has to select the right representative for that entity – George W. Bush (junior) or George H. W. Bush (senior) that will be probably both present in the ontology.

Relation Extraction is a task that usually comes after entity recognition. When all the significant entities are identified, the task is to connect together those entities that are connected in text and to assign the correct label (relation name) to that connection. For example let us have a text stating that George W. Bush is a son of George H. W. Bush. The extracted relation should be connecting the two entities (in the right order) and the label would be something like: "sonOf", "hasParent", "hasSon" or "hasChild" (depending on the system vocabulary and granularity; note the dependency between the label and the relation orientation.)

Event Extraction is similar to Relation Extraction but instead of connecting individual entities to simple relations, we are looking for richer events that are expressed in text. Individual events are labelled with proper event labels and entities are linked to corresponding events in proper roles. For example an *acquisition* event can have roles like *purchaser*, *seller*, *acquired*, *dollar amount*, etc.

3 Deep Linguistic Parsing and Information Extraction

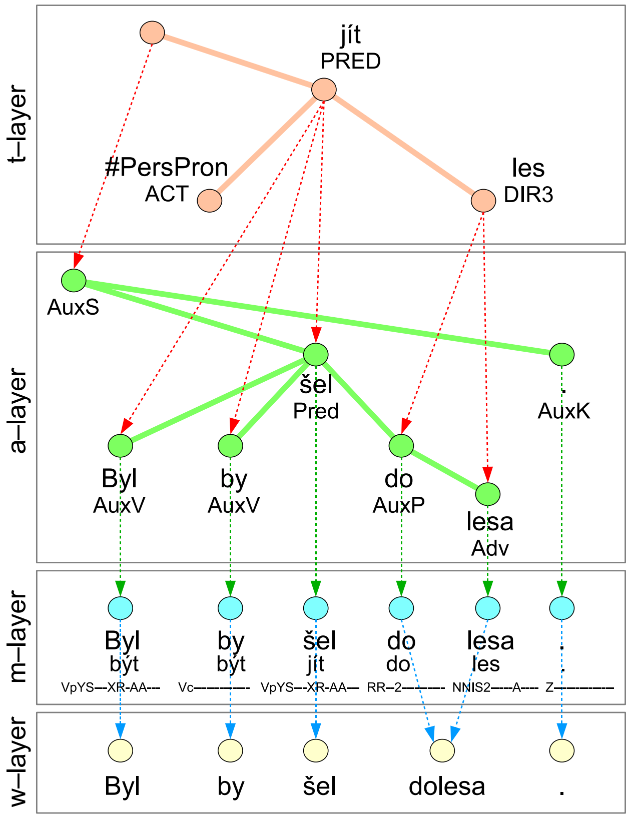
Dependency graphs constitute the fundamental data structure for syntactic structuring and subsequent knowledge extraction from natural language documents in many contemporary approaches to information extraction (see details about individual information extraction systems in Section 3.2). They use dependency structures generated by a variety of parsers. Besides the possibility of using different parsers there are also various language dependency representations (LDR) such as the Stanford [10] and CoNLL-X [7] dependencies and the Functional Generative Description (FGD) [13]. All the dependency representations are very similar from the structural point of view; they differ mainly in the number of different dependency kinds they offer. FGD provides also additional node attributes and so called layered approach (see in Section 3.1) and is the only representation available for Czech.

It is also worthy to investigate the impact of usage of different LDRs on the performance of information extraction. The authors of [3] compared the impact of different LDRs and parsers in their IE system, namely *Stanford* and *CoNLL-X* dependencies and usage of different trimming operations. Their findings are definitely not supporting one choice against another because one representation was better for some tasks and another one for other tasks, see examples in [3]. More important seems to be the quality of used parser.

The rest of this section provides a brief description of FGD, a basic overview of LDR based information extraction systems including our information extraction system based on FGD. Our aim is to investigate the possibility of using these technologies within the Khresmoi project, which is discussed in the last part of this section.

3.1 Functional Generative Description of Language

FGD provides a formal framework for natural language representation. It is viewed as a system of layers (stratification) expressing the relations of forms and functions. Although it was designed manly for the purpose of tree-based (dependency-based) machine translation (see e.g. in [16]), it can be used also for information extraction (details will be presented in Section 3.2.3) and our attention within the task T4.1b of the Khresmoi project was to investigate this possibility more closely, in the context of biomedical domain and the project needs.



Sample sentence (in Czech): Byl by šel dolesa.
English translation (lit.): [He] would have gone intoforest.

Figure 1 Layers of annotation in PDT.

Currently, FGD is practically realized on a number of corpora or treebanks such as the Prague Dependency Treebank[1] (PDT). There are three layers of annotation in PDT; see Figure 1 – a schema taken from PDT documentation[2], often used for the illustration of PDT annotation layers. We can start the description at the bottom of the picture: There is the surface structure of the sentence (w-layer) which represent the raw text without any modifications and annotations and is not considered as a layer of annotation (it can even preserve spelling errors, see e.g. the missing space between words 'do' and 'lesa' (into forest) in the example sentence, Figure 1). The lowest layer of annotation (m-layer) represents morphology (any spelling errors must be corrected at m-layer). Word forms are disambiguated into correct lemmas (dictionary form) and morphological tags are assigned at this layer.

[1] http://ufal.mff.cuni.cz/pdt2.0/

[2] http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/en/html/ch02.html

The second layer of annotation is called analytical (a-layer). At this layer the syntactical dependencies of words are captured (e.g. subject, predicate, object, attribute, adverbial, etc.) The top layer of annotation is called tectogrammatical (t-layer), sometimes also called “layer of deep syntax”. At this layer some tokens (e.g. tokens without lexical meaning) are leaved out or “merged’’ together (e.g. prepositions are merged with referred words). Also some nodes without a morphological layer counterpart are inserted to the tectogrammatical tree at certain occasions (e.g. a node representing omitted subject – in Figure 3.1 labeled as “#PersPron’’).

One important thing about t-layer is that it is designed as unambiguous from the viewpoint of language meaning [13], which means that synonymous sentences should have the same single representation on t-layer. Obviously this property is very beneficial to information extraction because t-layer provides certain generalization of synonymous phrases and extraction techniques do not have to handle so many irregularities.

Moreover all the layers are interconnected and additional linguistic features are assigned to tree nodes. Among others: morphological tag and lemma are assigned at the morphological layer; analytical function (the actual kind of particular analytical dependency, e.g. predicate, subject, object, etc.) is assigned to dependent nodes at the analytical layer; semantic parts of speech and grammatemes (e.g. definite quantificational semantic noun (n.quant.def), number, gender, etc.) and tectogrammatical functor (e.g. actor (ACT), patient (PAT), addressee (ADDR), temporal: when (TWHEN), directional: to (DIR3), etc.) are assigned at the tectogrammatical layer.

3.2 IE Systems Based on Deep Language Parsing

We describe in this section several information extraction systems based on deep language parsing. The systems differ greatly in the manner of using LDR.

3.2.1 Rule Based Systems

There are many systems using hand crafted extraction rules based on LDR. These systems need assistance from a human expert that is able to design the extraction rules manually. The advantage of these systems is that there is no need of learning (or training) data collection. For example in [8] a simple set of rules and the Stanford parser[3] were used for biomedical relation extraction. Shallow and deep parsers were used in [15] in combination with mapping rules from linguistic expressions to biomedical events.

There are also systems using some inductive technique e.g. Inductive Logic Programming (ILP) to induce extraction rules automatically from learning collection. In these cases it is not necessary to have a human rule designer but manual effort is needed for the construction of the learning collection. For example in [14], the dependency parser MINIPAR[4] and ILP were used for event extraction.

3.2.2 Classical Machine Learning Systems

Classical machine learning (ML) approaches rely on the existence of a learning collection. They usually use LDR just for construction of learning features for propositional learners like decision trees, neural networks, support vector machines (SVM), etc. Learning features are selected manually when the system is being adapted to new domain or extraction task. For example in [1], learning features based on so called dependency paths constructed from LDR are used for relation extraction. Similar approach was used in [2] for biomedical event extraction.

[3] http://nlp.stanford.edu/software/lex-parser.shtml

[4] http://webdocs.cs.ualberta.ca/~lindek/minipar.htm

There are also some hybrid systems performing automated induction of learning features, which is very similar to the induction of extraction rules; e.g. in [14], learning features for SVM classifier were induced using ILP.

3.2.3 CUNI's Extraction System based on Deep Language Parsing

The system was designed for vent extraction task. It relies on tree pattern extraction rules, that can be either designed manually [5] or induced from a learning collection [4]. In the later case ILP is used for the induction of extraction rules. Using ILP is beneficial as it is not necessary to transform dependency trees to some "simpler" propositional presentation: ILP can handle such multi-relational data naturally and the induced rules have then the same form as manually designed ones. Therefore the induced extraction rules can be easily visualized, understood and adapted by human.

Our information extraction system based on deep language parsing [4],[5] is available for the Khresmoi project. It employs several linguistic tools: tokeniser, sentence splitter, morphological analyzers (including stemmer and POS tagger), syntactic parser and deep syntactic (tectogrammatical) parser. The system is still in the development phase, but it has already been evaluated on the Acquisitions dataset[5] [9] and on one Czech proprietary dataset about car accidents; see details in [5]. Both the datasets represent an easy kind of event extraction task.

The system is not dependent on any particular dependency presentation. Some experiments were made with Stanford dependencies but the native presentation of the system is FGD, the only available representation for Czech.

3.3 Suitability of Deep Language Parsing to Different IE Tasks

Deep language models are very suitable for capturing the context of candidate entities that are being extracted. As already briefly mentioned in the previous section, it can be exploited in following extraction tasks:

Event extraction: Extraction rules and ML-based extraction models can discover new entities and determine correct entity roles based on the indications from the LDR representation of context ([2],[5] and [14]).

Relation extraction: Many relation extraction approaches exploit properties of the path through LDR structure that connects two candidate entities ([1] and an event extraction approach [2]).

Entity recognition: There are also successful works that can be seen as entity extraction and named entity extraction (e.g. [12]) exploiting LDR, but these works do not perform instance resolution. In these cases, entities (bits of text) are classified to small number of classes and the correct class is determined from the context directly.

The use of LDR for the **instance resolution task** is very limited. All the systems mentioned in this and the previous section were designed for relation or event extraction. During instance resolution, the entities have to be linked with corresponding ontology identifiers (e.g. MeSH IDs), which means that an extraction technique has to specify one of a very long list of all possible identifiers. Classical context oriented approaches make true/false decisions. They decide if a candidate entity should or should not be extracted. In such extraction system there is at least one extra extraction rule or ML-model for every entity kind. For a very long list of all possible identifiers (entity kinds) it would be necessary to have a very long list of extraction rules or ML-models. Such system would require huge amount of training data or huge effort to construct all the rules and the performance would probably

[5] Available from: http://nlp.shef.ac.uk/dot.kom/resources.html (2006-12-31)

not be much better than a conventional gazetteer based approach[6], which was finally selected as the Czech IE prototype, see Section 5.

3.4 LDR Adaption of Gazetteer Lookup

In this section, we will discuss one particular possibility how to employ LDR in the prototype. It is a LDR adaptation of the conventional gazetteer based approach. Instead of using just the sentence surface structure, gazetteer entries can be looked up using LDR. Gazetteer entries would have to be transformed to LDR patterns and matched against the language dependency structures of individual documents.

This approach would have the same advantage as the compound term analysis described in Section 5.1.2 because LDR structures are organized by dependencies, not by sentence word order, and therefore different word order of a gazetteer entry and word order of a term occurrence in text would not mind.

This approach would also improve the performance in cases when one or more words are inserted between words of a compound (multiword) gazetteer entry. For example instead of "patient's shoulder fracture" (*shoulder fracture* is the gazetteer entry) there could be "fracture of patient's shoulder" in text; in Czech the second formulation is even more frequent. Because the close connection of words of a compound term should look the same in both cases in LDR (for FGD this is one of the key properties), the pattern would match in both cases. The present example would look like a small tree with *shoulder* as the root node with two descendants: *patient* and *fracture* connect by the same dependencies in both cases. These cases are not covered by the compound term analysis but they could be probably very effectively solved by some kind of proximity search as well (without the necessity of deep language parsing).

This approach would be probably slightly more successful than the conventional approach with compound term analysis. Higher precision could be expected but the performance would be strongly influenced by the performance of used parser. From the evaluation results (Section 6.2) can be seen that the improvement made by compound term analysis is very small, similarly small improvement can be expected in this case and the cost of deep language parsing would decrease the time efficiency of the system.

3.5 Discussion about IE Tasks of the Project

In the Khresmoi project, information extraction is oriented mainly to information retrieval purposes. For example the recent use case documents (see the upcoming deliverable D8.2) provide an illustrative description of the project goals. Although the project annotation guidelines (see the upcoming deliverable D1.1) are currently in the finishing stage, they already provide information extraction tasks definitions. The main IE tasks of the project will be:

- document classification and document metadata extraction,
- structural annotation (e.g. identification of different document sections and their types) and
- annotation of biomedical terms (or entities).

[6] Gazetteers provide a list of known entities for a particular category, such as all counties of the world or all human diseases and are often used in information extraction. See also: http://gate.ac.uk/userguide/chap:gazetteers

Neither relation extraction nor event extraction (tasks that are frequently solved using LDR approaches) are covered by the guidelines and they do not seem to contribute to the project goals. The most promising candidate for improvement by LDR is the last point: annotation of biomedical terms as it is a kind of entity extraction. There are at least two good reasons why it is necessary to perform instance resolution as a part of the extraction task:

- The Khresmoi project is based on semantic technologies. Semantic indexing implemented in MIMIR (Multi-paradigm Information Management Index and Repository) provides a mapping of annotated (or extracted) terms to the project knowledge base. This is very useful e.g. for semantic queries profiting from all the interconnected knowledge. Such semantic indexing is only possible if the extracted entities are linked to corresponding unique identifiers.
- From the multilingual point of view, linking to unique identifiers provides a direct way to language neutral querying for documents because the unique identifiers are common for all languages. For example MeSH ID *D012784* corresponds with English term *Shoulder Fractures* but a query based on *D012784* instead of "Shoulder Fractures" will return documents in all available language, not only the English ones.

As discussed in the previous section, entity recognition accompanied with instance resolution can have only very limited prospect from using LDR; connected with the present discussion it lead us to the conclusion not to use LDR in our information extraction prototype.

4 Czech Biomedical Document Collection

This section provides description of a document collection that was used for two purposes:

1. as a database of Czech biomedical documents that were indexed within the project and
2. as an evaluation collection for measuring the performance of the Czech IE prototype.

The collection is based on a well established live project Bibliographia Medica Čechoslovaca (BMČ). BMČ is a Czech national register of biomedical and healthcare literature since 1947. The provided database (version 2011-01) contains 617,155 bibliographical entries (article references). BMČ entries refer to very strictly selected articles for medical specialists. For each article it contains a rich set of manually assigned metadata including several MeSH term annotations. Unlike, for example, PubMed[7] database, BMČ database does not contain abstracts of articles.

The collection is built up from documents publicly available on the internet. 25,408 available URLs were obtained from the BMČ database and corresponding documents were downloaded. But the URLs lead to external and heterogeneous destinations not always containing the content of referenced articles; sometimes the content was not readable at all; sometimes the character encoding was distorted. Therefore additional filtering was necessary to reach higher quality. To filter such heterogeneous collection we used a heuristics based on the MeSH terms density in text. Each document was quickly analyzed using a Czech MeSH gazetteer and documents where the density was lower than certain threshold (10 terms per 1 KB of text) were filtered out. This way reasonable quality was reached (random choice verification was done), but the collection was reduced to 11,410 files. See also the deliverable D4.1.1.

For the evaluation purpose, all available metadata from BMČ was kept with the documents. Sometimes several BMČ entries share the same URL and therefore the same target document (e.g. different articles from the same proceedings or journal issue), because we wanted to preserve the BMČ metadata along with the documents, merging of the BMČ entries was necessary. This has the effect

[7] http://www.ncbi.nlm.nih.gov/pubmed/

that metadata from more than one BMČ entry was assigned to some of the documents (e.g. to a document containing the whole proceedings or journal issue).

Table 1 provides detailed statistics of the document collection. Mean value, standard deviation, minimum, maximum and median values are available for all the statistics. The collection contains 11,410 documents of various lengths (from 10 to almost one million of characters ~ one to about 200,000 tokens respectively). The gold standard collection provides approximately 7.5 unique manually assigned MeSH terms per document; more than half of documents have at least 6 assigned MeSH terms (the median is 6); merging operations of BMČ entries were performed such that there is approximately 1.4 entry per document and more than half of the documents correspond to a single entry (the median is 1). The number of MeSH terms with separator character is relevant for compound analysis; the rate is quite high, there are approximately 4.7 MeSH terms with separator per document in the gold standard collection.

	mean	stdev	Min	max	median
Text length (characters):	15706.7669	28160.8748	10	966768	9714.5
Tokens:	3143.2286	5696.3840	1	203096	1947
Merged BMČ entries:	1.4379	5.0681	1	278	1
Unique MeSH terms:	7.4999	9.7138	0	426	6
Un. M. terms with separator:	4.7121	6.2072	0	250	4

Table 1: Gold standard collection (BMČ) basic statistics per document

5 Information Extraction Prototype for Czech

This section provides description of the Czech information extraction prototype for the Khresmoi project. It is based on the Czech translation MeSH taxonomy and it performs automated annotation of MeSH terms in text (entity recognition and instance resolution). Usually this is done using gazetteer lookup and statistical disambiguation (e.g. in [6]). For the Czech prototype there is no list of biomedical abbreviations or similar resource available, the MeSH vocabulary is not ambiguous and ordinary ambiguity of words is solved by morphological tagger (see bellow). Therefore the used extraction method does not contain any additional disambiguation procedure and is rather straightforward.

Before information extraction techniques can be applied to a document, linguistic analysis is performed as a preprocessing step. The linguistic analysis consists of sentence splitting, tokenization and morphological analysis. Tools developed at ÚFAL (CUNI) were used for these tasks. Morphological analysis is done by the Czech morphological tagger Morče[8] which provides also lemmatization.

The information extraction phase consists in medical named entity recognition and is based on the MeSH taxonomy. A gazetteer list was built from the MeSH taxonomy. As Czech is a flexitive language it cannot be used directly: term lookup without lemmatization would result in poor performance. Using GATE Flexible Gazetteer[9] this problem can be elegantly solved and terms from gazetteer list are then matched against tokens lemmas instead of their original forms. This also implies

[8] http://ufal.mff.cuni.cz/morce/

[9] http://gate.ac.uk/userguide/sec:gazetteers:flexgazetteer

that the gazetteer's terms have to be in the form of lemmas; therefore morphological analysis was performed on the gazetteer list itself during its construction.

5.1 Lemmatization and Compound Term Analysis

5.1.1 Lemmatization

The difference between lookup for Czech MeSH terms with and without use of lemmatization was investigated and Table 2 provides the final results on the whole dataset described in Section 4. The first column with four numbers shows the difference between the "Plain" setting (no use of lemmatization) and "Lemma" setting (use of lemmatization, but no compound analysis performed). The numbers represent mutual cross coverage of discovered MeSH terms in the following sense: only 0.3134 of terms found with lemmatization were also found without lemmatization and 0.9632 of terms found without lemmatization were also found with lemmatization. The bottom row provides the same when counting only unique terms for whole documents.

It is clear that lemmatization significantly improves the amount of found MeSH terms; however it is not 100% errorless. Closer investigation of the results showed that typical errors of the morphological tagger were uncovered, e.g. the phrase "otitis media" was tagged as "otitis medium" during the analysis of the gazetteer list and as "otitis medio" during the analysis of document. Section 6.2 provides statistics about how successful were these settings when compared with gold standard annotations.

	Plain	⇄	Lemma		⇄	Compound 1		⇄	Compound 2
All	0.3134	⇄	0.9632	0.9657	⇄	0.9987	0.9972	⇄	1
Unique	0.3991	⇄	0.9894	0.9634	⇄	1	0.9959	⇄	1

Table 2: Cross-coverage of extraction results (mean values)

5.1.2 Compound Term Analysis

When investigating Czech compound (multiword) MeSH terms it is soon quite obvious that great amount of them is not suitable for automated term search in text. It is because they are not in the form that one would expect to find in common text. Especially hyphen delimited terms (e.g. "Shoulder – Dislocation", "Shoulder – Fractures") are a dictionary form but this form is absolutely not common in text.

Even in English it is occasionally possible to write a compound term in different forms, e.g. "shoulder fracture" or "fracture of shoulder". Czech is a language with free word order and compound terms can be expressed in quite various ways. Therefore compound (MeSH) term analysis was included to the Czech information extraction prototype.

The compound term analysis consists in extension of gazetteer list by adding new items representing different permutations of suitable compound terms. It is done in three steps:

1. selection of suitable compound terms,
2. omission of delimiters (hyphens, commas, brackets) and
3. permutation of the remaining tokens.

The prototype with compound term analysis was tested in two settings:

Compound 1: only 2-token compound terms permutations were included and

Compound 2: 2-token and 3-token compound terms permutations were included.

For example 2-token compound term (with delimiter) “Shoulder – Fractures” was expanded to “Shoulder Fractures” and “Fractures [of] Shoulder”. In Czech, this is written as “rameno fraktury” and “fraktury rameno” and, thanks to the lemmatization, the second form is matching with the common phrase “fraktura ramene”. Similarly “Depressed Fracture [of] Skull” is reached as the sixth permutation ([3, 2, 1]) of the 3-token term “Skull Fracture, Depressed”.

The number of permutations is rapidly growing with the number of tokens; therefore it is not meaningful to apply this brute force method to longer terms. Table 3 gives details on the gazetteer extension. Compound 1 setting added one new permutation for every suitable 2-token term and Compound 2 setting added 5 new permutations for every suitable 3-token (not taking omission of delimiters into account). Suitable terms were selected as terms without prepositions and conjunctions (more precisely not containing shorter tokens than four charters) because permutation in these cases would not make much sense and it could lead to false positive matches.

Table 2 above shows also the mutual cross coverage in these settings. It can be seen that the more complex settings almost always cover the results of simpler settings and they always provide certain improvement; although the improvement of Compound 1 setting is very small (3-4%) and the improvement of Compound 2 is barely noticeable (less than 1%).

Table 4 shows numbers of compound term permutations that were found in the evaluation document collection. It is worth noting that permutations, where the order of two consecutive tokens was preserved ([2, 3, 1] and [3, 1, 2]), were more common than the others.

Section 6.2 provides statistics about how successful were these settings when compared with gold standard annotations.

Prototype setting	Number of entries
Plain & Lemma	26142
Compound 1	36776
Compound 2	54612

Table 3: Numbers of gazetteer entries in different settings of the IE prototype

2-token terms		3-token terms			
[1, 2]	158465	**[1, 2, 3]**	10402	**[2, 3, 1]**	1586
[2, 1]	50431	**[1, 3, 2]**	151	**[3, 1, 2]**	2660
		[2, 1, 3]	513	**[3, 2, 1]**	821

Table 4: Numbers of compound term permutations found in the document collection

6 Evaluation

Evaluation is an essential part of any scientific information extraction work. Evaluation provides a comparison of the results of a particular extraction method with the ideal results performed by human. Handcrafted gold standard data has to be obtained and the difference between the gold standard and extracted data is quantified by information extraction performance measures. Precision, recall and F1 (harmonic mean of precision and recall) are the mostly used ones.

Obtaining or building new gold standard data is costly and out of scope of this deliverable. New gold standard data for information extraction are often created within various benchmark projects and competitions such as ACE[10], BioCreative[11], etc., but multilingual information extraction benchmarking is still rather an unknown topic[12]. Because there is no multilingual biomedical IE benchmark available let us look for benchmarks for individual languages separately.

The multilingual information extraction prototype of the project can be seen as a composition of three sub systems:

1. The main IE system for English developed within the WP1 work-package.
2. MARVIN/WRAPIN, HON's medical multilingual service.
3. IE system for Czech based on GATE and CUNI's linguistic tools.

For other languages than English the situation is not very optimistic. For example CiSMeF[13] offers French biomedical corpus annotated by MeSH terms but we are able to have only a very limited access to CiSMeF, not enough for evaluation purposes.

There will be a lot of effort dedicated to evaluation of English information extraction in this and following years and it will be reflected in corresponding deliverables (D1.3, D1.8). Therefore we concentrate on non-English evaluation in this document.

The MARVIN/WRAPIN service is able to process documents in four non-English languages (French, German, Spanish and Portuguese). It is already a mature tool and it was partially evaluated previously (see the results presented in the previous multilingual deliverable: D4.1.1). But an exhaustive evaluation of the information extraction capabilities of the MARVIN/WRAPIN service is still lying ahead. It will be necessary to collect sufficient amount of multilingual documents accompanied with gold standard annotations. The project team is working in that direction and the first collection is being made for French using PubMed articles and manual annotations present on the PubMed site along with the corresponding articles. See the details about the data collection in [11]. The rest of this section deals with the evaluation of the MeSH term IE system for Czech described in Section 5 on the gold standard collection described in Section 4. Four experimental settings of the system are compared and the impact of the individual settings on the system performance is discussed.

6.1 MeSH Vocabulary Statistics

Several measurements were performed before the information extraction prototype was constructed. The analysis of the Czech MeSH vocabulary was an important precursor of the compound term analysis (see Section 5.1).

[10] http://www.itl.nist.gov/iad/mig/tests/ace/

[11] http://www.biocreative.org/

[12] One can imagine such benchmark as a collection of parallel texts in different languages with the same (language neutral) extraction tasks. In this case it would be possible to compare performance of different language modules with each other and it may also turn out that information extraction in one language is more difficult than in another. The requirement of parallel texts is optional. A multilingual benchmark could also be a collection of different texts in different languages with the same extraction tasks. Such benchmark would evaluate a multilingual extraction system as a whole, but it would be not possible to compare different language modules to each other reasonably.

[13] http://www.chu-rouen.fr/cismef/

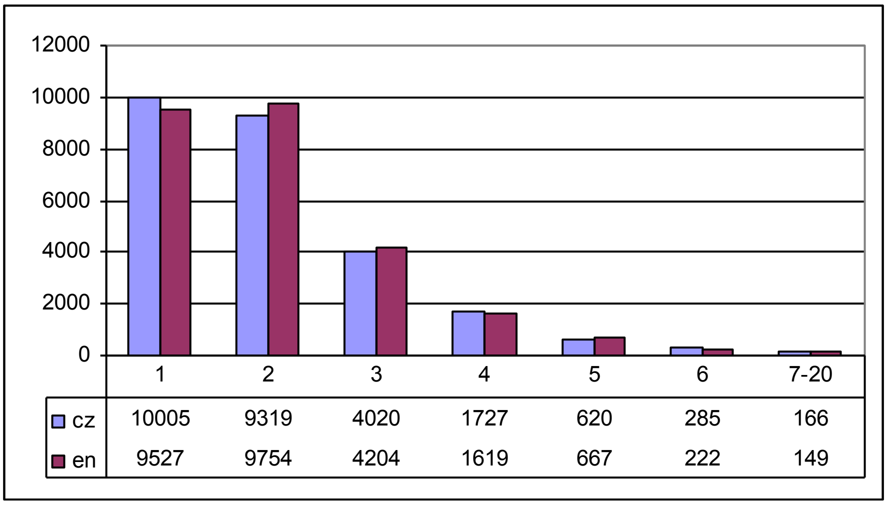
Frequency analysis of characters used in the Czech and English MeSH vocabulary was performed; see Table 5. Token separating characters were selected from the results. The table clearly shows high rate of hyphen usage and lower rate of commas in the Czech vocabulary.

Frequency analysis of numbers of MeSH terms tokens (frequencies of term lengths) provides an outlook to how much impact the compound analysis will have according to the token length of expanded MeSH terms; see Figure 2.

Finally, an interesting view of the vocabulary is provided by the results of frequency analysis of token lengths; see Figure 3. A slightly off-topic observation can be made that Czech tokens tend to be one character shorter than English ones (in the most frequent lengths).

	A-Z	0-9	space	( )	-	,	other
Czech	435607	2418	28871	296	**5518**	234	126
English	431326	2433	24769	224	2892	**4275**	195

Table 5: Character counts in Czech and English MeSH



	1	2	3	4	5	6	7-20
□ cz	10005	9319	4020	1727	620	285	166
■ en	9527	9754	4204	1619	667	222	149

Figure 2: Frequencies of Czech and English MeSH terms according to the number of tokens

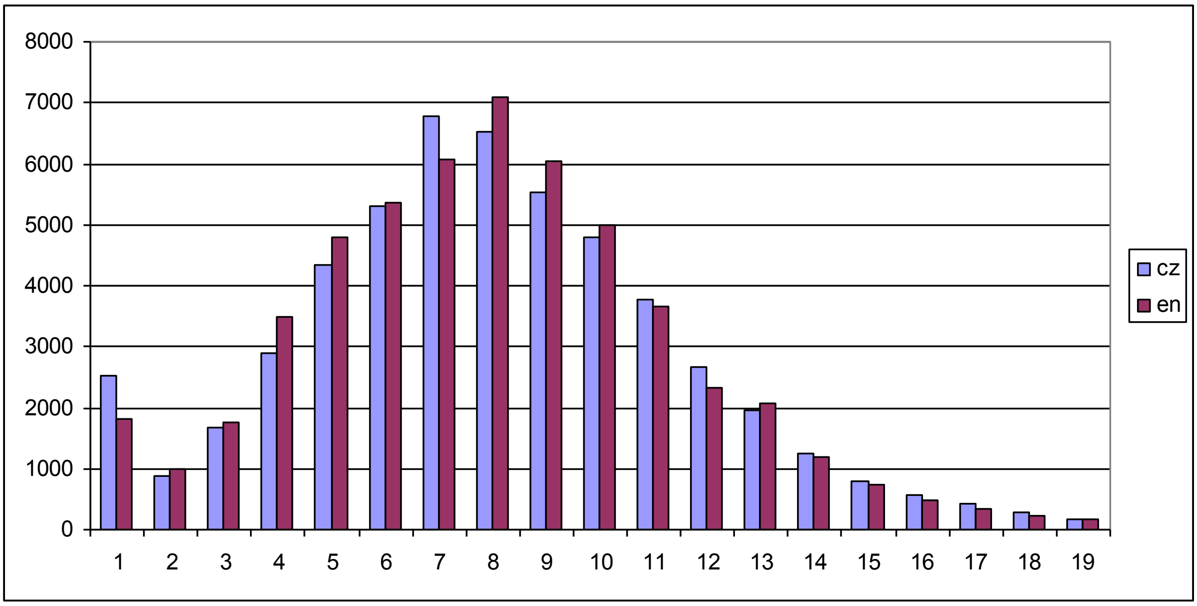


Figure 3: Frequencies of Czech and English MeSH term tokens according to token length

6.2 System Performance Statistics

Evaluation of the Czech IE prototype was performed in four different settings (see Section 5.1 for details). Table 6 provides quantitative comparison of number of terms discovered using these settings. The most obvious observation is that the number of discovered non unique terms far exceeds gold standard and, on the contrary, the number of unique discovered terms is only slightly higher than the number of unique gold standard terms.

Next interesting observation is that the number of 2-token terms is higher than the number of single token terms in gold standard and (vice versa) the number of sing-token terms is higher than the number of 2-token terms for discovered terms in all four settings. Another important finding relates to the difference between the number of 3-token terms discovered in Compound 1 and Compound 2 settings. This is exactly the improvement of 3-token compound analysis (the difference is 5,777 terms, which is 0.30% of all Compound 1 terms ~ 310 and 2.76% for unique terms).

	Gold Standard	Plain	Lemma	Compound 1	Compound 2
All	84,704	578,865	1,834,479	1,907,169	1,912,946
Unique	9,456	8,644	9,998	11,227	11,537
1 token	31,486	521,810	1,654,247	1,674,058	1,674,058
2 tokens	**35,723**	50,745	166,014	218,854	218,854
3 tokens	11,504	5,537	13,401	**13,436**	**19,213**
4 tokens	4,214	683	686	690	690
5-8 tokens	1777	90	131	131	131

Table 6: Number of gold standard and discovered MeSH terms in the whole collection

Table 7 summarizes evaluation of the prototype against BMČ gold standard MeSH terms annotations. Standard information extraction performance measures were used: precision, recall and F1. The values of precision (and hence also F1) are not to be considered serious for two reasons:

1. the gold standard annotations were not intended for IE evaluation by their authors and
2. the system was not optimized for high precision.

It is indeed not clear if the high precision is desirable in this case. If we look at user needs, the most important thing will most probably be the correct ranking of a document against user query and the more information a ranking function will have the better. Therefore all correctly discovered terms and their frequencies are valuable. The main issue with using BMČ for evaluation is that it cannot reveal incorrectly discovered terms because the fact that a term is not present in BMČ entry does not mean that the term is not relevant to the particular document.

Also values of recall are quite low. This is probably caused by lower quality of both: gold standard data and the document collection. Further cleaning of both would probably increase these numbers. It is also very likely that many of the gold standard terms are not explicitly mentioned in text. For example an article about dental caries susceptibility could be annotated with term “Dentistry” but such term does not need be explicitly mentioned in the text of that article. Development of an algorithm for “guessing” (or inference) these not mentioned terms is an interesting task for possible future work.

We think that, although the results are not really convincing in the absolute sense, they are very valuable for relative comparison of different settings of the prototype. It is clear that lemmatization brought improvement of recall, and both settings of compound analysis improved recall as well as precision. On the other hand the improvement made by compound analysis is not so significant and similarly low improvement can be expected from the usage of LDR.

From the statistical point of view the results are not significant (too high deviations) and paired comparison would be necessary to investigate the statistical significance. Also usage of BDM (balanced distance metric) would make the evaluation more adjusted because of the hierarchical character of MeSH taxonomy.

		Mean	stdev	Min	Max	median
Recall	Plain	0.2011	0.2978	0	1	0.0833
	Lemma	0.2903	0.2964	0	1	0.2000
	Compound 1	0.3109	0.3002	0	1	0.2500
	Compound 2	0.3137	0.3007	0	1	0.2500
Precision	Plain	0.0570	0.1295	0	1	0
	Lemma	0.0379	0.0652	0	1	0.0244
	Compound 1	0.0382	0.0615	0	1	0.0265
	Compound 2	0.0384	0.0616	0	1	0.0265
F1	Plain	0.0662	0.1255	0	1	0
	Lemma	0.0581	0.0842	0	1	0.0426
	Compound 1	0.0595	0.0801	0	1	0.0465
	Compound 2	0.0599	0.0803	0	1	0.0465

Table 7: Performance evaluation statistics per document

7 Conclusion

In this document, multilingual information extraction prototype of the Khresmoi system was introduced. Then the Czech module of the system was described in detail in order to investigate the possibility of using deep language models in the prototype. The issue was investigated from different perspectives and the conclusion ends up against the usage of deep language models in the prototype because it does not seem feasible. The main arguments can be summarized as follows.

Language dependency representation (LDR) is a useful means for information extraction, but not all kinds of extraction tasks can benefit the same from using it.

Two possibilities how LDR could be used within the project were discussed in this document:

1. Usage of CUNI's Extraction System based on Deep Language Parsing described in Section 3.2.3 and
2. An LDR Adaption of Gazetteer Lookup discussed in Section 3.4.

The first possibility is not suitable because this extraction system is designed for event extraction and, as discussed in Section 3.5, neither event extraction nor relation extraction are employed within the Khresmoi project.

The second possibility is very similar to the classical gazetteer based approach that was finally selected as the Czech information extraction prototype of the project. Based on discussion of this approach in Section 3.4 and the evaluation results of compound term analysis, it can be concluded that the potential improvement of the system would be rather low and the use of deep language parsing would decrease the time efficiency of the system.

8 References

[1] Bunescu, R., Mooney, R. Extracting relations from text: From word sequences to dependency paths. In: Kao, A., Poteet, S.R. (eds.) Natural Language Processing and Text Mining, chap. 3, pp. 29-44. Springer, London (2007)

[2] Buyko, E., Faessler, E., Wermter, J., Hahn, U.: Event extraction from trimmed dependency graphs. In: BioNLP '09: Proceedings of the Workshop on BioNLP. pp. 19-27. Association for Computational Linguistics, Morristown, NJ, USA (2009)

[3] Buyko, E., Hahn, U. Evaluating the impact of alternative dependency graph encodings on solving event extraction tasks. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 982-992. (2010)

[4] Dědek, J. Towards semantic annotation supported by dependency linguistics and ILP. In Proceedings of the 9th International Semantic Web Conference (ISWC2010), Part II, volume 6497 of Lecture Notes in Computer Science, pages 297-304, Shanghai / China, 2010. Springer-Verlag Berlin Heidelberg. ISBN 978-3-642-17748-4. URL http://iswc2010.semanticweb.org/accepted-papers/219. (2010)

[5] Dědek, J., Vojtáš, P. Exploitation of linguistic tools in semantic extraction - a design. In Mieczysław Kłopotek, Adam Przepiórkowski, Sławomir Wierzchoń, and Krzysztof Trojanowski, editors, Intelligent Information Systems XVI, pages 239-247, Zakopane, Poland, 2008. Academic Publishing House EXIT. ISBN 978-83-60434-44-4. URL http://iis.ipipan.waw.pl/2008/proceedings/iis08-23.pdf. (2008)

[6] Dill, S., Eiron, N., Gibson, D., Gruhl, Guha, D., R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J., A., Zien. J. Y. SemTag and seeker: bootstrapping the semantic web via automated semantic annotation. In Proceedings of the 12th international conference on World Wide Web (WWW '03). ACM, New York, NY, USA, 178-186. URL http://doi.acm.org/10.1145/775152.775178 (2003)

[7] Johansson, R., Nugues, P. Extended Constituent-to-dependency Conversion for English. In Proceedings of NODALIDA 2007, pp. 105–112, Tartu, Estonia. (2007)

[8] Fundel, K., Kuffner, R., Zimmer, R.: Relex - relation extraction using dependency parse trees. Bioinformatics 23(3), 365-371 (2007)

[9] Lewis, D., D.: Representation and learning in information retrieval, Ph.D. thesis, University of Massachusetts, (1992).

[10] De Marneffe, M., MacCartney, B., Manning, Ch., D. Generating 85 Typed Dependency Parses from Phrase Structure Parses. In Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology, The Stanford Natural Language Processing Group, URL http://nlp.stanford.edu/pubs/LREC06_dependencies.pdf. (2006)

[11] Park, J., Schneller, P. Building Automatically a MeSH annotated French Corpus. Technical Report ELDA-2012-02. (2012)

[12] Pecina, P. Lexical Association Measures: Collocation Extraction, volume 4 of Studies in Computational and Theoretical Linguistics. UFAL, Praha, Czech Republic, (2009)

[13] Sgall, P., Hajičová, E., Panevová, J. The Meaning of the Sentence in Its Semantic and Pragmatic Aspects, Dordrecht: Reidel Publishing Company and Prague: Academia, ISBN 978-90-277-1838-9. (1986)

[14] Ramakrishnan, G., Joshi, S., Balakrishnan, S., Srinivasan, A.: Using ILP to construct features for information extraction from semi-structured text. In: ILP'07: Proceedings of the 17th international conference on Inductive logic programming. pp. 211-224. Springer-Verlag, Berlin, Heidelberg (2008)

[15] Yakushiji, A., Tateisi, Y., Miyao, Y., Tsujii, J.: Event extraction from biomedical papers using a full parser. Pac Symp Biocomput pp. 408-419 (2001)

[16] Žabokrtský, Z., Ptáček, J., Pajas, P.: TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer, in Proceedings of the 3rd Workshop on Statistical Machine Translation, pp. 167–170, ACL, Columbus, OH, USA, ISBN 978-1-932432-09-1. (2008)