DELIVERABLE

Project Acronym: **EUCases**

Grant Agreement number: **611760**

Project Title: **European and National Legislation and Case Law Linked in Open Data Stack**

D3.10 Report on multilingual access

Authors:

Kiril Simov	IICT-BAS
Petya Osenova	IICT-BAS
Iliana Simova	IICT-BAS
Ivajlo Radev	IICT-BAS

Project co-funded by the European Commission within **FP7-ICT-2013-SME-DCA**		
Dissemination Level		
PU	Public	■
PP	Restricted to other programme participants (including the Commission Services	
RE	Restricted to a group specified by the consortium (including the Commission Services	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Revision History, Status, Abstract, Keywords, Statement of Originality

Revision History

Revision	Date	Author	Organisation	Description
0.1	01-04-2015	Petya Osenova	IICT-BAS	First draft
0.2	11-04-2015	Kiril Simov	IICT-BAS	Ontology-to-text relation
0.3	21-04-2015	Iliana Simova	IICT-BAS	Parallel corpora
0.4	22-04-2015	Ivajlo Radev	IICT-BAS	Extension of Bulgarian part of Eurovoc
0.5	30.04.2015	Kiril Simov	IICT-BAS	Ontology-based translation module
0.6	26-05-2015	Iliana Simova	IICT-BAS	Statistical Machine Translation module
1.0	29.05.2015	Kiril Simov, Petya Osenova	IICT-BAS	Formatting and Editing for final Submission

Date of delivery	Contractual:	30-04-2015	Actual:	31-05-2015
Status	final ☑/draft ☐			

Abstract (for dissemination)	This document presents the components of the multilingual access modules implemented within the project. We have developed two modules for translation of user queries between Bulgarian and English as demonstration for multilingual access modules. The goal of the task is to design and implement modules for ensuring translation of user queries written in the source language to the target language and retrieval of documents in the target language. We have implemented two such modules: (1) statistical machine translation module and (2) ontology-to-text translation module. The former exploits the progress in the area of machine translation. The latter is based on a common ontology with aligned lexicons, and reflects the areas of query expansion and ontological inference.
Keywords	Statistical Machine Translation, Ontology-to-Text relation, Parallel corpora, annotation grammars, Aligned lexicons

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of Content

Executive Summary

This deliverable introduces the EUCases Multilingual Access Module. This module translates the user's legislation query from its source language into the target language, and retrieves the detected texts that much the query. The service is demonstrated in its potential for two languages – English and Bulgarian, in both directions (English-to-Bulgarian and Bulgarian-to-English).

The module consists of two submodules: Ontology-based one and Statistical Machine Translation one.

The Ontology-based one relies on EuroVoc thesaurus, which provides aligned lexicons in all the EU official languages in legislation domain. This submodule uses the ontology-to-text relation approach, which provides a strategy for annotating text with concepts with the help of chunk grammars. It ensures the realization of query expansion. The user is provided with various control options for using or not the query expansion service.

The Statistical Machine Translation one uses parallel as well as monolingual data for creating language and translation models. It relies on Moses as a state-of-the-art translation system. This submodule incorporates two approaches: word-form-based and part-of-speech-tags-based. The English-to-Bulgarian direction uses wordform and part-of-speech tag as factors. The Bulgarian-to-English direction uses the same factors plus lemma.

Since both proposed submodules have drawbacks, they are used in an integrated architecture, thus profiting from each other.

1 The EUCases Multilingual Access Module

In this deliverable we report on the design and development of Multilingual Access Module (MLAM) for full text search within EUCases documents in English and Bulgarian. The MLAM translates user queries in both directions: from Bulgarian to English and from English to Bulgarian. A typical user query is a list of key words and phrases. The user query is evaluated over a set of documents loaded in a full text search engine which performs searches for relevant documents. In our case, the full text search engine is provided by APIS. Thus, our goal is to deliver an adequate translation service for the user queries.

In the module two complementary technologies are exploited. The first technology is based on Ontology-to-Text relation. In this case, the system relies on a common ontology with augmented lexicons. The lexicons are mapped in such a way that the conceptual information within the ontology corresponds to the meaning of the lexical items. Having lexicons for different languages aligned to the same ontology is a prerequisite for the accurate translation of the corresponding lexical items. In addition to the lexicons, special chunk grammars are needed to recognize the lexical items in the text. Such grammars are important especially for languages with rich morphology and/or free word order.

The exploitation of ontology in translation provides additional functionality of performing query expansion on the basis of inference within the ontology. In our module we implemented two query expansion procedures: (1) expansion via subclasses and (2) expansion via related classes. Both of them are presented in our ontologies: Syllabus ontology and Eurovoc multilingual taxonomy. After performing the query expansion, the new set of classes is translated to the target language using the appropriate lexicons.

Within the project we expect that the user queries will be mainly related to the above mentioned domain ontologies which are also employed for document indexing. They also reflect the specific content of the documents in the EUCases database. Nevertheless, the users should not be restricted to specify their queries only through the lexical items from the available lexicons. Thus, MLAM needs to provide translation also for words and phrases that are not in the lexicons. In order to solve this problem, we exploit a statistical machine translation module trained on domain specific parallel corpora. This module in combination with the ontology-based module provides alternative translations to the lexicon items and thus covers the missing translations for out-of-vocabulary words and phrases.

2 Ontology-Based Translation Module

The design and implementation of the Ontology-based translation module of MLAM exploits the ontology-to-text relation presented in deliverable D1.1 of the project. We started with classification of ontologies with respect to their precision provided by Nicola Guarino (Guarino 2000):

- **Lexicon**: machine readable dictionaries; vocabulary with natural language definitions.
- **Simple Taxonomy**: classifications.
- **Thesaurus**: WordNet; taxonomy plus related-terms.
- **Relational Model**: Light-weight ontologies; unconstrained use of arbitrary relations.
- **Fully Axiomatized Theory**: Heavy-weight ontologies.

The two ontologies to be used in the project - Syllabus ontology and Eurovoc multilingual taxonomy - are at the middle of the classification. The first one is a relational model by its creation, and the second one is a thesaurus. Thus, they provide a limited inference which we exploit for query expansion. Both of them are considered domain ontologies, but are not aligned to an upper ontology.

2.1 Ontology-to-text relation

Here we represent again the two main components that define the *ontology-to-text* relation necessary to support the tasks within our project. These components are: (terminological) lexicon and concept annotation grammar.

Lexicon plays a twofold role in our architecture. First, it interrelates the concepts in the ontology to the lexical knowledge used by the grammar for recognizing the role of the concepts in the text. Second, lexicon represents the main interface between the user and the ontology. This interface allows for the ontology to be navigated or represented in a natural for the user way. For example, the concepts and relations might be named with terms used by the users in their everyday activities and in their own natural language. This might be considered as first step to a contextualized usage of the ontology in a sense that the ontology could be viewed through different terms depending on the context.

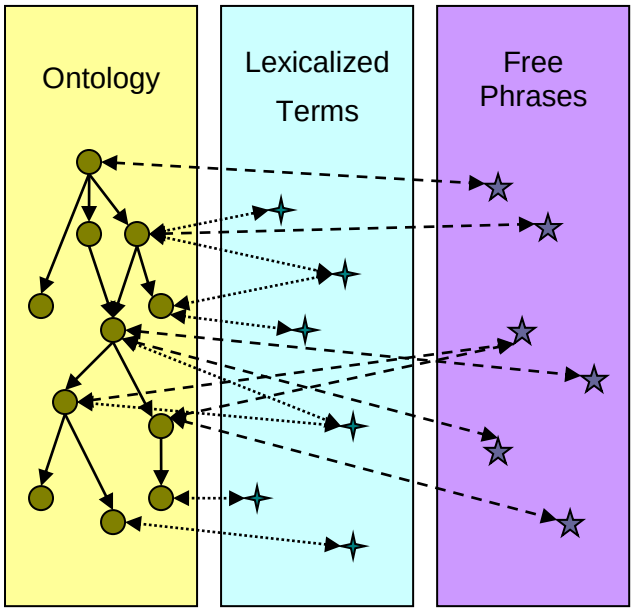
Thus, the lexical items contain the following information: a term, contextual information determining the context of the term usage, grammatical features determining the syntactic realization within the text. In the current implementation of the lexicons the contextual information is simplified to a list of a few types of users (lawyers, judges, etc).

With respect to the relations between the terms in the lexicon and the concepts in the ontology, there are two main problems: (1) there is no lexicalized term for some of the concepts in the ontology, and (2) there are lexical terms in the domain language which lack corresponding concepts in the ontology.

The first problem is solved by adding also non-lexicalized (fully compositional) phrases to the lexicon. These varieties of phrases or terms for a given concept are used as a basis for construction of the annotation grammar. Having them, we would capture different wordings of the same meaning in the text. The picture below shows the mapping varieties. It depicts the realization of the concepts (similarly for relations and instances) in the language. The concepts are language independent and they might be represented within a natural language as form(s) of a lexicalized term, or as a free phrase. In general, a concept might have a few terms connected to it and a (potentially) unlimited number of free phrases expressing this concept in the language. Some of the free phrases receive their meaning compositionally regardless their usage in the text, other free phrases denote the corresponding concept only in a particular context. In our lexicons we decided to register as many free phrases as possible in order to have better recall on the semantic annotation task. In case of a concept

that is not-lexicalized in a given language we require at least one free phrase to be provided for this concept.

Figure 1: ontology-to-text relation



We could summarize the connection between the ontology and the lexicons in the following way: the ontology represents the semantic knowledge in form of concepts and relations with appropriate axioms; and the lexicons represent the ways in which these concepts can be realized in texts in the corresponding languages. Of course, the ways in which a concept could be represented in the text are potentially infinite in number. For that reason we aimed at representing in our lexicons only the most frequent and important terms and phrases.

The second component of the ontology-to-text relation, the concept annotation grammar, is ideally considered as an extension of a general language deep grammar which is adopted to the concept annotation task. Minimally, the concept annotation grammar consists of a chunk grammar for concept annotation and (sense) disambiguation rules. The chunk grammar for each term in the lexicon contains at least one grammar rule for recognition of the term. As a pre-processing step we consider annotation with grammatical features and lemmatization of the text. The disambiguation rules exploit the local context in terms of grammatical features, semantic annotation and syntactic structure, and also the global context, such as topic of the text, discourse segmentation, etc.

For the implementation of the annotation grammar we rely on the grammar facilities of the CLaRK System (Simov et al., 2001). The structure of each grammar rule in CLaRK is defined by the following DTD fragment:

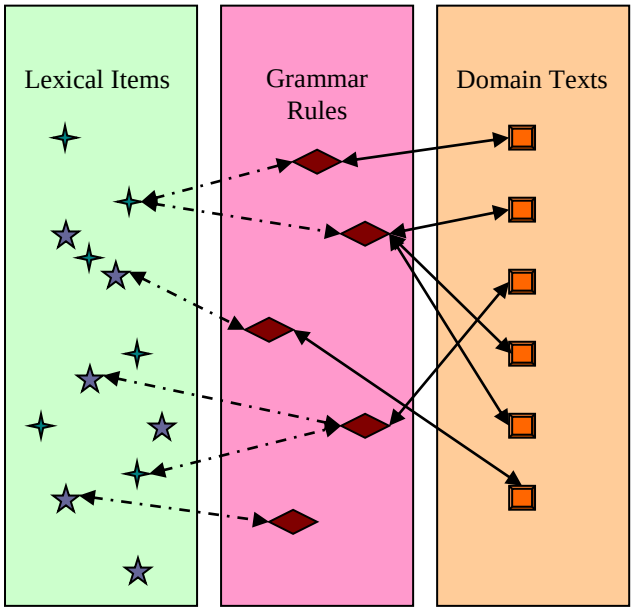
```
<!ELEMENT line (LC?, RE, RC?, RM, Comment?) >
<!ELEMENT LC (#PCDATA)>
<!ELEMENT RC (#PCDATA)>
<!ELEMENT RE (#PCDATA)>
<!ELEMENT RM (#PCDATA)>
<!ELEMENT Comment (#PCDATA)>
```

Each rule is represented as a line element. The rule consists of a regular expression (*RE*) and a category (*RM* = return markup). The regular expression is evaluated over the content of a given XML element and could recognize tokens and/or annotated data. The return markup is represented as an XML fragment which is substituted for the recognized part of the element content. Additionally, the user could use regular expressions to restrict the context in which the regular expression is evaluated successfully. The *LC* element contains a regular expression for the left context, and the *RC* - for the right one. The element Comment is for human use. The application of the grammar is governed by *Xpath* expressions which provide an additional mechanism for accurate annotation of a given XML document. Thus, the CLaRK grammar module is a good choice for implementation of the initial annotation grammar. The construction of the chunk grammar can be done also in other frameworks.

The creation of the actual annotation grammars started with the terms in the lexicons for the corresponding languages. Each term was lemmatized and the lemmatized form of the term was converted into a regular expression of grammar rules. Each concept related to the term is stored in the return markup of the corresponding rule. Thus, if a term is ambiguous, then the corresponding rule in the grammar contains a reference to all concepts related to the term.

The following picture depicts the relations between lexical items, grammar rules and the text:

Figure 2: relation between lexical items, grammar rules and text



The relations between the different elements of the models are as follows. A lexical item could have more than one grammar rule associated to it depending on the word order and the grammatical realization of the lexical item. Two lexical items might share a grammar rule if they have the same wording, but are connected to different concepts in the ontology. Each grammar rule recognizes zero or several text chunks.

The ontology-to-text relation, implemented in this way, provides facilities for solving different tasks, such as ontology search (including crosslingual search).

2.2 Implementation of Ontology-Based Translation Module

Our first implementation of the ontology-based translation module used EuroVoc[1] multilingual taxonomy as its main ontology. We consider EuroVoc as a light-weight ontology. EuroVoc covers the concepts related to the European Union activities. In this respect we consider it as a domain ontology for the main topics of interest to the users of EUCases project services. The lexicons were aligned within EuroVoc for the official languages of European Union and also some other languages. In this respect, it provides the necessary inventory for the two languages that are important to us: Bulgarian and English. It also can be extended further for the other represented languages. The actual concepts are mapped via numerical identifiers. The concepts are arranged in domains and microthesauri. Each domain is divided into a number of microthesauri. A microthesaurus is considered as a concept scheme with a subset of the concepts that are part of the complete EuroVoc thesaurus. The main relations encoded in EuroVoc that we exploit are: "skos:broader"[2], "skos:related" and "xl:prefLabel", defined in http://www.w3.org/2009/08/skos-reference/skos.html and http://www.w3.org/TR/skos-reference/skos-xl.html documents. Here are a few examples of a concept in EuroVoc:

Concept ID	Bulgarian term	English term	skos:narrower	skos:related
1460	Финансов инструмент на Общността	EU financial instrument	1052, 2054, 2609	862, 1851, 2511, 4370, 5472
1052	Фондове на ЕС	EC fund	5138, 5643, 978	973, 4055, 8549, 862
5138	Структурни фондове	Structural funds	1056, 4056, 5668	5472, 5499, 5580, 5847
862	Помощ на Общността	EU aid	852	-
5499	Икономическо и социално взаимодействие	Economic and social cohesion	5864	5643
5643	Фонд за сближаване	Cohesion Fund	-	-

As it was mentioned above, on the basis of all terms related to a given concept identifier, a regular expression is created. Each rule annotated the recognized text with the corresponding identifier. In the cases of ambiguous terms, the text was annotated with several identifiers. The annotation grammars were applied over the user query string and each recognized term was presented by the corresponding concept identifiers. After the complete analysis of the input query, the text was converted into a list of identifiers. In some cases, a specific substring of the user query might not be recognized as a term in the lexicon. In this case, the substring remained unanalysed. For example, for the concept 1460 the grammar rule in CLaRK system would look like:

<"EU"> , <"financial"> , <"instrument"> → <concept v="1460"/>

where the first part of the rule would recognize the term in the text and this text would be annotated with the XML fragment from the right part of the rule.

[1] http://eurovoc.europa.eu/

[2] Also its reverse relation " skos:narrower "

After the user query was excecuted, the text was annotated with concept identifiers. Then we performed ***query expansion*** on the basis of the ontology. In this case, we exploited the two relations that define the structure of the ontology: "skos:narrower" and "skos:related". As the table shows, both relations "skos: narrower" and "skos:related" are transitive. These relations can be used for adding new concept identifiers to those from the annotation of the user query. The relations can be used in two ways: (1) getting only the directly related concepts, or (2) getting the concepts that are related via transitive closure of the relations. In the first implementation, we performed a transitive closure for the relation skos:broader and only direct related concepts for the relation skos:related. Here we present the different steps of processing of the user query: EU financial instrument:

Step 1: Text annotation

EU financial instrument → <concept v="1460"/>

Step 2: Query expansion applying transitive closure of skos:narrower

<concept v="1460"/> →

<concept v="1460"/><concept v="1052"/><concept v="5138"/><concept v="1056"/>

<concept v="4056"/><concept v="5668"/><concept v="980"/><concept v="5643"/>

<concept v="978"/><concept v="979"/><concept v="2054"/><concept v="2609"/>

<concept v="2607"/><concept v="2608"/><concept v="2610"/><concept v="738"/>

Step 3: Query expansion applying skos:related

<concept v="1460"/><concept v="1052"/><concept v="5138"/><concept v="1056"/>

<concept v="4056"/><concept v="5668"/><concept v="980"/><concept v="5643"/>

<concept v="978"/><concept v="979"/><concept v="2054"/><concept v="2609"/>

<concept v="2607"/><concept v="2608"/><concept v="2610"/><concept v="738"/>

→

<concept v="1460"/><concept v="862"/><concept v="1851"/><concept v="2511"/>

<concept v="4370"/><concept v="5472"/><concept v="1052"/><concept v="973"/>

<concept v="4055"/><concept v="8549"/><concept v="862"/><concept v="5138"/>

<concept v="5472"/><concept v="5499"/><concept v="5580"/><concept v="5847"/>

<concept v="1056"/><concept v="976"/><concept v="2407"/><concept v="2516"/>

<concept v="6061"/><concept v="4056"/><concept v="5668"/><concept v="980"/>

<concept v="5643"/><concept v="978"/><concept v="979"/><concept v="2054"/>

<concept v="4838"/><concept v="2609"/><concept v="862"/><concept v="4370"/>

<concept v="2607"/><concept v="4838"/><concept v="5675"/><concept v="2608"/>

<concept v="852"/><concept v="2610"/><concept v="5344"/><concept v="738"/>

<concept v="739"/>

Step 4: Deleting the repeated concepts

The above result contains repeated concepts which were deleted before the actual substitution of the concept identifiers with the term in the other language. The result is:

<concept v="1460"/><concept v="862"/><concept v="1851"/><concept v="2511"/>

<concept v="4370"/><concept v="5472"/><concept v="1052"/><concept v="973"/>

<concept v="4055"/><concept v="8549"/><concept v="5138"/><concept v="5499"/>

<concept v="5580"/><concept v="5847"/><concept v="1056"/><concept v="976"/>

<concept v="2407"/><concept v="2516"/><concept v="6061"/><concept v="4056"/>

<concept v="5668"/><concept v="980"/><concept v="5643"/><concept v="978"/>

<concept v="979"/><concept v="2054"/><concept v="4838"/><concept v="2609"/>

<concept v="2607"/><concept v="5675"/><concept v="2608"/><concept v="852"/>

<concept v="2610"/><concept v="5344"/><concept v="738"/><concept v="739"/>

Step 5: Translation to the other language

In this step each concept identifier is substituted with the corresponding terms in the other language. The result for our example is:

> финансов инструмент на ЕС помощ на ЕС поддържащ механизъм земеделска валутна политика европейска парична система рамка за подкрепа на общността фондове (ЕС) европейски фонд за валутно сътрудничество европейски фонд за развитие европейски фонд за приспособяване към глобализацията структурни фондове икономическо и социално взаимодействие структурен разход подходящ район за развитие европейски фонд за регионално развитие регионална помощ регионално планиране регионална политика на ЕС структурно приспособяване европейски социален фонд фиор европейски фонд за ориентиране и гарантиране на земеделието секция „ориентиране“ фонд за сближаване ФЕОГА европейски фонд за ориентиране и гарантиране на земеделието - секция „гарантиране“ нов инструмент на общността европейска инвестиционна банка заем от общността заем на европейска инвестиционна банка инициатива за европейски растеж заем на европейско обединение за въглища и стомана помощ на европейско обединение за въглища и стомана заем на ЕВРАТОМ ЕВРАТОМ заем, получен от ЕС международен заем

This translation of the expanded query is used for retrieval of the appropriate documents from the full text search system.

The approach for query expansion is based on the intuition that when someone searches for a concept they are interested in all subconcepts of the given one as well as related concepts with step one from the initial concept because the related concepts that are far from the initial concept could introduce too much unrelated content. In order to provide more flexible control over the query expansion we have implemented the following combinations:

- NQE: No query expansion
- QNA: Query expansion using transitive closure of the relation skos:narrower
- QRE: Query expansion using the relation skos:relate
- QNR: Query expansion using both relations

The implementation provides a possibility for the user to select the translation direction: Bulgarian-to-English, English-to-Bulgarian as well as the query expansion approach: one of the above.

As one can see, there are many ways for query expansion. The best one would depend on the domain, the task and so on. After the evaluation of this module we will improve it to achieve a better performance.

3 Statistical Machine Translation Module

As it was mentioned above, our task is to handle the out-of-vocabulary items for the lexicons aligned to the ontologies, and also to provide a module for translation of user queries based on statistical machine translation. User queries are mainly lists of key words and phrases which we expect to be domain dependent. Thus the parallel corpora on which the Statistical Machine Translation system (SMT) to be trained are very important.

As a system for statistical machine translation we selected Moses[3]. Moses is a data-driven and state-of-the-art machine translation system. It provides three types of translation model implementations: phrase-based models, where n-grams ("phrases") are the basic units of translation, hierarchical or syntax-based models, where information about the structure of the parallel data can be incorporated, and factored translation models, where additional linguistic information (e.g., lemma, part-of-speech tag) can be integrated into the translation process.

Moses has two main components – a training pipeline and a decoder. The training pipeline includes various tools for data pre-processing (e.g., for tokenisation, lowercasing, removing very long sentences on the source or target side of the training data, etc.), for accessing external tools for data alignment (GIZA++), language model building (SRILM, KenLM, IRSTLM, RandLM), and implementations of popular tuning algorithms. The Moses decoder tries to find the highest scoring sentence during translation, or outputs a ranked list of translation candidates with additional information. Standard tools for the evaluation of translations (e.g., BLEU scorer) are also available.

In addition to parallel data in the form of plain text, Moses can be used to decode data represented as confusion networks or word lattices. In this way, ambiguous input data, such as the output of an automatic speech recognizer, or a morphological analyzer, can be processed to reduce erroneous hypothesis.

Thus the machine translation systems for the language pairs English-Bulgarian and Bulgarian-English were created using the Moses open source toolkit (see Koehn et. al. (2007)). Parallel data from several sources was used to train factored translation models (Koehn and Hoang, 2007), which can be viewed as an extension to standard phrase-based models, where more linguistic information can be utilized in the translation process in addition to word forms.

3.1 Used Corpora

The following data sets were used for creating the translation models of the SMT system:

3.1.1 Parallel Corpora

We used several parallel corpora:

SETimes[4] (154K sentences)

A parallel corpus of news articles in the Balkan languages, originally extracted from http://www.setimes.com. Here we are using Bulgarian-English part which was cleaned within European EuroMatrixPlus project[5]. We manually checked the alignment for more than 25000 sentences. The rest was automatically cleaned from sentence pairs that were suspicious with respect to their translation.

[3] http://www.statmt.org/moses/

[4] http://opus.lingfil.uu.se/SETIMES.php

[5] http://www.bultreebank.org/EMP/

Europarl[6] (380K sentences)

The parallel texts were extracted from the Proceedings of the European Parliament.

Bulgarian-English BTB lexicon (9K word translations)

Lexicon created by professional lexicographers especially for machine translation purposes.

JRC Acquis[7] (364K sentences)

Parallel texts were extracted from the European Union legislation.

EAC-ECDC[8] (7K sentences)

These sentences were extracted from translation memories published on the above web page. The sentences were extracted manually because of the many alignment discrepancies.

APIS Legal Corpus (3844 sentences)

These sentences were extracted from parallel texts covering part of Bulgarian legislation.

The parallel data was cleaned semi-automatically. Non-translated sentences in the Bulgarian data were detected and removed together with their equivalents in the English data. Empty sentences or sentences longer than 80 words were removed with the Moses script clean-corpus-n.perl. The parallel data was lowercased for training.

In the parallel data corpora we used mainly domain related corpora, but we also included an out-of-domain corpus in order to cover also more general language usage.

3.1.2 Monolingual Corpora

The following data sets were also used for the creation of suitable language models:

- Bulgarian: National Reference Corpus (1.4M sentences) and the Bulgarian data from the parallel corpora
- English: Europarl (2M sentences) and the English data (without Europarl) from the parallel corpora

3.2 Data Analysis

The basic units of translation in Factored translation models are vectors of factors, where a factor is usually a linguistic property attached to word forms, such as the lemma, part-of-speech tag, or morphological properties associated with the word. In the current settings, we make use of the linguistic analyses produced by the Bulgarian pipeline btb-pipe (implemented partially within EUCases project) and the English system ixa-pipes (see (Agerri et. al 2014)). In the preprocessing step, we perform sentence splitting, tokenization, part of speech tagging, and lemmatization of the Bulgarian and English parallel corpora.

The Bulgarian data was processed with the pipeline btb-pipe, which includes rule-based, hybrid, and statistical components. It performs tokenization and sentence splitting jointly. The tagging module assigns tags from a rich tagset (Simov et. al 2004), which encodes detailed information about the morphosyntactic properties of each word. In the current experiment, simplified POS tags were used as factors for translation to avoid data sparsness issues. Lemmatization is based on rules, generated with the help of a morphological lexicon.

[6] http://www.statmt.org/europarl/

[7] http://optima.jrc.it/Acquis/index_2.2.html

[8] https://ec.europa.eu/jrc/en/language-technologies/ecdc-translation-memory

For the processing the English part of the data we created a wrapper for the modules of the *ixa-pipes* system (Agerri et. al 2014) *ixa-pipe-tok* (version 1.7.0), and *ixa-pipe-pos* (version 1.3.3). The wrapper includes an additional module which generates factored output, suitable for use with the Moses factored system. The first module, *ixa-pipe-tok*, takes plain text input, and carries out rule-based tokenization and sentence segmentation. The next step in the pipeline, *ixa-pipe-pos*, includes POS tagging and lemmatization. For tagging we selected one of the provided POS models for English - Perceptron (Collins 2002). The wrapper provides the option to preserve the number of lines in the input and output English file. This option should be used in the processing of parallel corpora to ensure that the resulting factored output can be aligned to its corresponding Bulgarian file in case when the English file contains more than one sentence on certain lines.

3.3 Translation Models

During training, a mapping between the source and target language linguistic factors produced in the previous step is established. The following lines describe the alternatives which were chosen for the two translation directions of the English and Bulgarian languages.

The SRI Language Modeling Toolkit (SRILM) Koehn et. al. (2007) was used to build 5-gram language models for the two translation systems. Two types of language models were used by both systems – word-form-based and part-of-speech-tags-based ones. For the word-based language models we use Kneser-Ney smoothing, while part-of-speech-tag-based models were smoothed with the Witten-Bell method.

Both systems were tuned with minimum error rate training (MERT) (see (Och 2003)) implementation provided as part of the Moses toolkit.

The English-to-Bulgarian translation system uses two types of linguistic factors - wordform and part-of-speech tag. The wordform of the source language is mapped to wordform and POS tag factors on the target language side. Additionally, the proposed wordform and POS tag sequences are evaluated with the corresponding language models.

The Bulgarian-to-English system uses three types of factors - wordform, lemma, and part of speech tag. In this translation scenario a source word form is again mapped to target word form and tag, but if no translation is found for the word, its lemma is used instead. This greatly helps with data sparseness issues caused by the rich morphology of Bulgarian. Once again two language models were employed, one for tags and one for wordforms.

The translation process can be divided into the following steps:

1. The input is processed with the corresponding pipeline, and factors are generated.
2. The factored data is lowercased (Moses script lowercase.perl).
3. The factored data is translated with the corresponding model.
4. The translation output is de-tokenized (Moses script detokenizer.perl).
5. The translation output is re-cased (a Moses re-caser model was trained for each translation direction).

3.4 Statistical Translation Web Services

The translation web service was built with mt-monkey (https://ufal.mff.cuni.cz/mtmonkey) see (Tamchyna et. al 2013).

Usage examples:

#en-bg:

curl -i -H "Content-Type: application/json" -X POST -d '{"action":"translate", "sourceLang":"en", "targetLang":"bg", "text": "Hello, world!" }' http://213.191.204.69:9990/btb

#bg-en

curl -i -H "Content-Type: application/json" -X POST -d '{"action":"translate", "sourceLang":"bg", "targetLang":"en", "text": "Здравей, свят!" }' http://213.191.204.69:9990/btb

The translation modules used in the implementation of these web services are also integrated with the module for ontology-based translation in order to implement a joint web service.

4 Integration of Two Modules

Both translation modules have some drawbacks. The ontology-based translation module is not able to disambiguate ambiguous terms because of the lack of annotated corpora from where a statistical model to be learnt. It also cannot translate out-of-vocabulary text. Such out-of-vocabulary text might have new paraphrases of existing concepts in the ontology or complete unrelated keywords or phrases. The statistical machine translation module also might not be able to use enough contexts in the user search query to translate some keywords or phrases in the right way. Also, the SMT module is not able to translate words that are not mentioned in the training corpus, thus sometimes the translation contains words in the source language. In order to handle these drawbacks of both modules and to gain from their complementary performance, we ingrate them in the following way:

First, we translate the user query Q by each of the modules. The results are Qot and Qsmt. Each of them could contain substrings from the source language. We delete them from the two translated queries. Then we concatenate the two strings. The result is Qtrans. This result is used for the full text search in a document database in the target language.

The integration module is available via a web service which provides options for the various query expansion approaches.

Usage examples:

#en-bg:

curl -i -H "Content-Type: application/json" -X POST -d '{"action":"translate", "sourceLang":"en", "targetLang":"bg", "queryExpansion" : "QNR", "text": " EU financial instrument " }'

http://213.191.204.69:9990/uqt

#bg-en

curl -i -H "Content-Type: application/json" -X POST -d '{"action":"translate", "sourceLang":"bg", "targetLang":"en", "queryExpansion" : "QNA", "text": "финансов инструмент на ЕС" }'

http://213.191.204.69:9990/uqt

References

Rodrigo Agerri, Josu Bermudez and German Rigau. 2014. IXA pipeline: Efficient and Ready to Use Multilingual NLP tools. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14).

Collins, Michael. 2002. Discriminative training methods for Hidden Markov Models. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10. pp 1-8.

Philipp Koehn and Hieu Hoang, 2007. Factored Translation Models. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 868-876.

Koehn, Philipp and Hoang, Hieu and Birch, Alexandra and Callison-Burch, Chris and Federico, Marcello and Bertoldi, Nicola and Cowan, Brooke and Shen, Wade and Moran, Christine and Zens, Richard and Dyer, Chris and Bojar, Ondřej and Constantin, Alexandra and Herbst, Evan. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. pp 177-180.

Och, Franz Josef. 2003. Minimum Error Rate Training in Statistical Machine Translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. pp. 160-167.

Kiril Simov, Petya Osenova, and Milena Slavcheva. 2004. BTB-TR03: BulTreeBank morphosyntactic tagset BTB-TS version 2.0.

Aleš Tamchyna, Ondřej Dušek, Rudolf Rosa, and Pavel Pecina. 2013. MTMonkey: A scalable infrastructure for a Machine Translation web service. In Prague Bulletin of Mathematical Linguistics 100, 2013, pp. 31-40.