UDK: 004.9:811.16’374
811.16'322

i-Librarian – Free online library for European citizens

Diman Karagiozov (diman@tetra.com), Tetracom Consulting
Anelia Belogay (anelia@tetracom.com), Tetracom Consulting
Dan Cristea (dcristea@info.uaic.ro), Alexandru Ioan Cuza University
Svetla Koeva (svetla@dcl.bas.bg), Institute for Bulgarian Language, Bulgarian Academy of Sciences
Maciej Ogrodniczuk (maciej.ogrodniczuk@ipipan.waw.pl), Institute of Computer Science, Polish Academy of Sciences
Polivios Raxis (raxis@atlantisresearch.gr), Atlantis Consulting
Emil Stoyanov (emil@tetracom.com), Tetracom Consulting
Cristina Vertan (cristina.vertan@uni-hamburg.de), University of Hamburg

Abstract

The emergence of the WWW as the main source of distributing content opened the floodgates of information. The sheer volume and diversity of this content necessitate an approach that will reinvent the way it is analysed. The quantitative route to processing information which relies on content management tools provides structural analysis. The challenge we address is to evolve from the process of streamlining data to a level of understanding that assigns value to content.

The solution we present incorporates human language technologies in the process of multilingual web content management. i-Librarian is a website built with an open-source software platform ATLAS. ATLAS complements a content management software-as-a-service component used for creating, running and managing dynamic content-driven websites with a linguistic platform. The platform enriches the content of these websites with revealing details and reduces the manual work of classification editors by automatically categorising content. The platform supports six European languages – Bulgarian, German, Greek, English, Polish, and Romanian.

i-Librarian is a free online library that assists authors, students, young researchers, scholars, librarians and executives to easily create, organize and publish various types of documents. It allows users to maintain their personal workspaces for storing, sharing and publishing various types of documents and have them automatically categorized, summarized and annotated with important words, phrases and names. It also allows

similar documents in different languages to be found easily.

Keywords
digital library, language processing chains, UIMA, content management system, multilinguality, machine translation, summarization, classification

Introduction

The advent of the Web revolutionized the way in which content is manipulated and delivered. As a result, digital content in various languages has become widely available on the Internet and its sheer volume and language diversity have presented an opportunity for embracing new methods and tools for content creation and distribution. Although significant improvements have been made in the field of web content management lately, there is still a growing demand for online content services that incorporate language-based technology.

The existing software solutions and services such as Google Docs, Slingshot and Amazon implement some of the linguistic mechanisms addressed in the platform. The most used open-source multilingual web content management systems (Joomla, Joom!Fish, TYPO3, Drupal)[1] offer low level of multilingual content management, providing mechanisms for building multilingual sites. However, the available services are narrowly focused on meeting the needs of very specific target groups, thus leaving unanswered the rising demand for a comprehensive solution for multilingual content management addressing the issues posed by the growing family of languages spoken within the EU.

We are going to demonstrate the open source content management platform and, as a proof of concept, a multilingual library driven by the platform. The paper aims to prove that people reading websites powered by our multilingual web management platform can easily find documents, kept in order via the automatic classification, find context-sensitive content, find similar documents in a massive multilingual data collection, and get short summaries in different languages that help the users to discern essential information with unparalleled clarity.

1. The online library

i-Librarian is a free online library that assists authors, students, young researchers, scholars, librarians, executives to easily create, organize and publish various types of documents and share them with other people at no cost. i-Librarian reduces the manual work of classification editors by using the automatic classification of editions, news, documents and provides summarizes of documents and their translations. Moreover, the users can easily find the most essential texts from

1 http://www.joomla.org/, http://www.joomfish.net/, http://typo3.org/ http://drupal.org/

large document collections and can receive better publication overview due to the extended information published based on text annotation: valuable names, dates, numerical expressions and important phrases.

The i-Librarian system is developed within ATLAS (Applied Technology for Language-Aided CMS) – a project funded by the European Commission under the CIP ICT Policy Support Programme; Grant Agreement 250467) [2]. Its main purpose is to facilitate the multilingual Web content development and management, in particular the authoring, versioning and maintenance of multilingual Web sites. One of the project's main achievements is i-Librarian which gains by the integration of language technologies in the multilingual content management.

The basic functionality of the system, from the point of view of different types of users, is described in the following subsections.

1.1. Reader's corner

Executives, academics, people who travel, people who enjoy reading:

1. A reader uploads several digital books to their personal workspace. The books are processed by the service, organized into appropriate subject categories, summarized, and annotated with important words and phrases. The reader can then access and read the books from anywhere in the world with a browser or a mobile device (iPhone, Android-based devices, etc.). Furthermore, they can discuss their favourite books with other users of i-Librarian, and if the reader particularly likes a book, they can search for similar books regardless of the language.
2. An executive has a business meeting with a potential customer out of the office. They want to show an important document to the client but the document is not available on their portable memory. They access their i-Librarian account and easily find the required document because all documents are categorized, with extracted summaries.

1.2. Author's corner

Students, researchers, analysts, consultants:

1. A student is writing a research paper and needs to quickly select and read the most essential texts from a large collection of reports, news articles and scientific publications. They upload an archive with all documents to their i-Librarian workspace. The service summarizes the documents and extracts important words, phrases and names. After reading the summaries and text extracts, the student decides which documents are worth taking a closer look at and which could be discarded. Furthermore, they can easily navigate in their workspace because i-Librarian automatically assigns uploaded files to appropriate subject categories and interlinks documents based on text extracts. Finally, the student can publish the completed paper either to the i-Librarian public section or to existing web sites.
2. A researcher publishes a paper in i-Librarian. The service automatically annotates the paper with important words, phrases and names, and translates the annotations into several languages. Another researcher who works in the same field but speaks a different language finds the document with the i-Librarian "find similar" multilingual search and contacts the author. As a result, the two researchers can share their knowledge on a particular subject and choose to work together.
3. A scientist gives a speech at a scientific forum on their current research results.

2 http://www.ATLASproject.eu

After the lecture, a question from the audience is raised and the scientist has to support the answer with a fact published in a conference paper. They access the i-Librarian account and find the paper, stored under the "Conferences" leaf of the categorization tree. As a user of the i-Librarian service, they had the opportunity to upload all documents, papers, research publications and, using the clustering functionality, to organize and keep them in order.

1.3. Librarian's corner

1. A digital edition needs to be represented in an appealing way in order to get the readers' attention. The bookstore uses the i-Publisher service to process the digital content and enrich the available information for the digital edition of the book. In addition to the bibliographic information like author, title and date of publication, every edition comes with a summary generated by the i-Publisher so that the reader can get a quick overview of the book content. The reader gets most frequently used noun phrases, names, links, and dates for this book. Clicking on a phrase, for example, the reader finds the list of books in which this phrase is featured. The reader is presented with a list of digital books that are similar to the one currently viewed.
   Benefit: Adding value to content. The system leads the user to content relevant to the one he is initially interested in. In addition, the user easily makes the choice of books. The bookstore will capitalise on extended book sales since users find it easier to locate relevant information i.e. find books on very specific topics. It will also benefit from the bulk sales that will be increased by the suggested similar documents, enabling readers to find and purchase multiple books on their favorite topics.
2. A library publishes vast amounts of information like publications, books, articles and bulletins, etc. which have to be annotated, categorised and made available online on a daily basis. The library integrates the Text Mining Tools in the existing software system. A team member trains a model for the categorisation of the digital content using manually categorised data or integrates a pre-trained model. As a result newly added content will be automatically categorised according to that model. The newly added content is enriched with automatically compiled annotations such as extraction of the most-commonly used noun phrases in the text, dates, links, named entities and a detailed extractive summary. In addition, the annotations are machine-translated in the languages available for the website.
   Benefit: Automatic categorisation. Manual work done by classification editors will be reduced as the system automatically suggests categories for the content items. The additional information published on the website gives the user a better publication overview. In addition, the suggested list of similar documents can be very useful in finding relevant information on a topic.

2. Creating the library

The i-Librarian web application is developed as a demonstration of the functionalities of a novel content management system called ATLAS **(Ogrodniczuk and Karagiozov 2011)**. Being a content management system (CMS), ATLAS was used for configuring the data model of i-Librarian, its look-and-feel, user registration and profiles, maintenance of isolated private user space. Furthermore, the ATLAS CMS provides means of advanced content editing, configurable content approval workflows, and granular access

rights system, flexible. look-and-feel configuration, rich selection of predefined themes and content models.

An easy to use graphical user interface is built on top of the ATLAS CMS core. Based on ZK,[3] ATLAS graphical user interface (GUI) is a cross-browser, scalable and secure, rich internet application (RIA).

As an application developed entirely with ATLAS, i-Librarian uses techniques that enable intelligent processing of information and add value that can't be delivered by other means. The following list of features shortly describes the key techniques and algorithms in ATLAS that are used for building intelligent web applications:

- Indexing and full text searching – a modern CMS allows the information designers to structure the content and the relations between the content items dynamically, and later to run full-text search queries in the pool of content items. The most widely used full-text search engine library that is integrated in CMS is Apache Lucene or tools based on Lucene, such as Apache Solr.
- Identification of important "cue" words and phrases – nouns (and noun phrases) are traditionally defined as "persons, places, things, and ideas". Amazon first defines the term "Statistically improbable phrases" as "the most distinctive phrases in the text of a particular book … relative to all books (in a collection)". The main added value to a CMS is the presentation to the user of the main concepts and ideas of a content item.
- Identification of named entities – the named entities are noun phrases which are further disambiguated and categorized by their meaning and function in the text. The extracted named entities are used for answering the 5W1H questions (who, what, why, where, when and how) and for finding similar content. Popular services providing NE extraction are OpenCalais, Stanford CoreNLP and OpenNLP.
- Clustering similar content items – filtering, reviewing and maintaining the relations between the content items is time and effort consuming task for the information designers and content providers. Thus, a CMS needs tools which provide functionalities like "more like this", "recommended reading", and "see also". According to the cluster hypothesis ("Documents in the same cluster behave similarly with respect to relevance to information needs.") the most significant features of content item are almost the same in similar content items from one and the same cluster.
- Automatic assignment of tags to the content items – tagging the content (assigning keywords) facilitates its searching and finding; however, the process requires a lot of manual efforts. Taxonomy building and tags assignments are two techniques that can be performed semi-automatically by the computers and reviewed and corrected manually.
- Computer aided translation for multilingual web applications – being a thriving research field, machine translation (MT) is a new functionality, poorly integrated in the process of content management. On the other hand, the demand for multilingual web sites is rapidly increasing. The MT engines assist the content providers with the initial translation of textual materials; they also help the web application users to cross the language barriers. The existing services providing MT are Moses, Google Translate, Bing Translator.

3 http://www.zkoss.org/

3. Behind the digital shelves

This chapter of the article elaborates on the details of ATLAS architecture and natural language processing component which is the core of the features, specific for the intelligent web applications. According to our knowledge, there is currently no content management system that transparently integrates and provides easy-to-use interface to natural language processing tools. The aim of ATLAS CMS is to enable integration of the exiting heterogeneous NLP tools in the process of content management.

3.1. Language processing chains

Textual information is generally unstructured; however, humans are able to process it and find the most important pieces of it. Computers, on the contrary, cannot perform such analysis – they are programmed to execute a sequence of tasks in order to reveal the main concepts and interrelations in the text. The sequential tasks, called a language processing chain (LPC), consist of atomic NLP tools which add low-level annotations in the text and thus make it structured. We use the low-level annotations to extract important words and phrases, and named entities at a later stage of processing. Furthermore, we apply statistical algorithms to the low-level annotations in order to find the most significant features of the analysed text.

A sample LPC consist of the following atomic NLP tools: Tokenizer (splits the raw text into tokens) → Paragraph splitter (splits the text in paragraphs) → Sentence splitter (splits the paragraphs in sentences) → POS tagger (marks up each token with its particular part of speech tag) → Lemmatizer (determines the lemma for each token) → Word sense disambiguation (disambiguates each token and assigns an unique sense to it) → NP Extractor (marks up the noun phrase in the text) → NE Extractor (marks up named entities in the text) (see Figure 1).

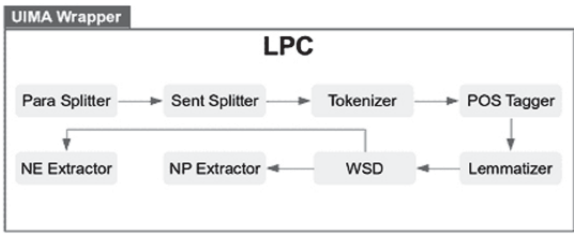

Figure 1. Sample Language Processing Chain.

In order to achieve optimal precision of the LPC, we combine statistics-based NLP tools with language specific linguistic rules. For instance, the English LPC consists of the following components, executed in a sequence:

1. Paragraph splitter and URL/Email annotator – based on a set of regular expressions.
2. Sentence splitter, Tokenizer and PoS tagger based on OpenNLP.
3. Lemmatizer – uses the Morphological Analysis tool from the RASP (Robust Accurate Statistical Parsing) system (v. 2).
4. Noun phrase extractor – the grammar and structure of the English noun phrase are described in a set of 14 rules, following the format of ParseEst sub-component.
5. Named entities recognizer – based on seven NE models from OpenNLP for recognition of dates, time expressions, locations, money expressions, organization names, percentage and person names.

3.2. ATLAS requirements and architecture

The major non-functional requirements for the CMS are:

- Responsiveness – the classical request-response scenario should be performed as fast as possible.
- Scalability – the CMS should scale horizontally and vertically in order to achieve maximum performance.
- Maintainability – a CMS is rich of major and minor functionalities which are

often overlapping and/or complimentary. The maintenance of a CMS is not a trivial task, thus the architecture should support this process as much as possible.

- Inter-operability – the interface between a CMS and other systems should be as standard as possible. This will allow future extensions and integration of external functionalities.

The integration of any language processing modules must not compromise any of these four major requirements. Here is how we address each of the above-listed non-functional requirements:

- Responsiveness. Usually, the NLP tasks are slow. Their overall performance depends on performance of the atomic NLP tools and the size of the input text. This is the reason why a language processing chain cannot be instantiated in the classical request-response chain because response time cannot be predicted. Thus, we are using an asynchronous communication channel between the CMS and LPC components.
- The CMS asynchronously sends a message, identifying the document and providing its content to the LPC engine and informs the user that the request is being processed. The appropriate status of the task is shown to the user while the message is being processed by the LPC engine. The results of the task become available in the CMS once the message is eventually processed.
- OSGi-based Language Processing Chains engine. The OSGi framework is a module system and service platform for Java that implements a complete and dynamic component model. Applications or components can be remotely started, stopped, and updated without requiring a reboot. Equinox, an OSGi framework implementation, has been chosen as a backbone of suggested LPC engine architecture. Our architecture consists of three main components:
  - Message queue. Java Messaging Service (JMS) API is a message oriented middleware for sending messages between two or more clients. It allows the communication between different components of a distributed application to be loosely coupled, reliable, and asynchronous. We have based the implementation of the transport messaging agent, between the CMS and the different LPC components on the Apache ActiveMQ.
  - Atomic annotator. The atomic annotator is responsible for the initial set of annotations needed by the higher NLP tasks. The annotator checks-out a message from the queue and delegates the processing to:
    - Pre-processor. The component identifies the mime-type of the message content, extracts the text if needed, detects the language of the text and sends an internal message to the NLP processor;
    - NLP processor. The component provides the basic annotations in the message text. Similar to the OSGi for Java, UIMA (Unstructured Information Management Applications) allows the complex NLP applications to be decomposed into components. Each atomic NLP tool is wrapped into UIMA primitive engine; the primitive engines are sequenced by an aggregate engine. UIMA is not OSGi compliant, thus we wrapped the UIMA aggregate engine in an OSGi component (NLP processor), making it available to the rest of the components in the

installation;
    - Post-processor. The component is invoked when the annotations are ready, saves the annotations, provides performance report, informs the CMS that the annotations are available, and invokes the higher-level categorizer and summarizer components.
  - Categorizer and Summarizer. The categorizer and the summarizer have one and the same internal architecture, thus, only the summarizer is described in details. The component checks-out a message from a queue, loads the needed sentences and tokens for the requested document, instantiates a summarization engine (LexRank implementation or OpenText Summarizer external tool), creates a summary of the document and sends the summary to a queue to be further processing (saved in a data store).
- Scalability. The usage of message queue in the architecture of the LPC engine enables a trivial horizontal scalability by simply installing new instances of the LPC engines. Typical i-Librarian users are not English native speakers, thus the expectation is that they will use i-Librarian in bi-lingual environment – native and English languages. Thus, the expected number of English documents is much greater than the number of documents in other languages. It makes sense, in this case, to deploy several English LPCs working in parallel in order to minimize the time prior to a message being processed.
- The architecture described above is depicted on the diagram in Figure 2.

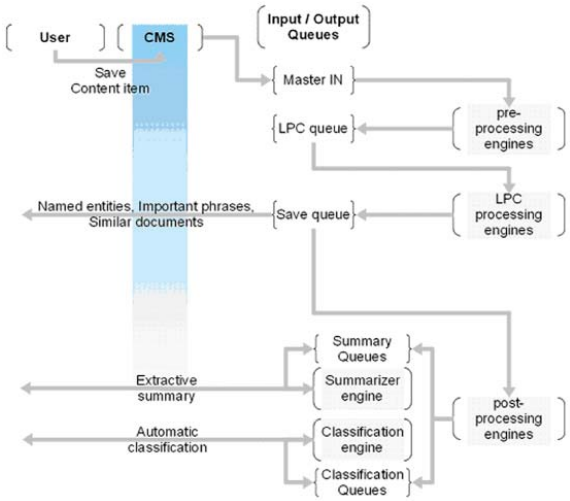

Figure 2. ATLAS major architectural components and communication channels between them.

The block on the diagram reads:

- User. The users trigger the language processing chains by manipulating (adding or updating) the content on a web site.
- Content management system. The CMS "communicates" with the LPC engines via message queue through a well-defined API. Currently, only OSGi-based API is available.
- Input & Output queues. The asynchronous communication between the components is empowered by a JMS implementation. A message is sent to an input queue; a component checks-out the message, transforms it and sends it to another queue. The LPC component and the CMS implement the message router, message translator, messaging gateway, event-driven consumer and competing consumers enterprise design patterns (Hohpe and Woolf, 2003)
- Pre-processing engines. The component provides mime-type detection, text extraction, language identification and text cleanup.
- LPC processing engines. The component

wraps a LPC for a given language.
- Post-processing engines. The components store the annotations in a data store.
- Summarization and Categorization engines. These components provide a summary and list of categories applicable to a document. The architecture of the engines allows the integration of multiple summarization algorithms and categorization tools.

The ATLAS architecture, being based on OSGi specification, can be easily extended to support more languages (currently Bulgarian, English, German, Greek, Polish and Romanian are available in the form of LPCs) and more types of annotations, such as co-reference chains and deeper integration with WordNets, in order to achieve better semantic understanding of the textual content.

4. Language technology for multilingual collections

Multilingual library content is processed by language-dependent processing chains which offer the same set of processing actions for all supported languages (paragraph and sentence boundary detection, tokenization, lemmatization, POS tagging, NP chunking, NE recognition). Even though the technical components applied to processing differ from language to language, this approach offers the common ground for language processing and its results can be comfortably used by advanced language components (content-based document classification, statistical machine translation, clause-based summarization) as well as for direct visualization.

4.1. Automatic categorization

Document classification is the task to assign a document to one or more categories or classes. Automating this process is of great importance for modern applications; therefore, a variety of methods have been developed during the years. The methods for automatic classification can be informally divided into two groups – statistical algorithms and structural algorithms. Examples for statistical algorithms are Regression and Naïve Bayes. Structural algorithms can be further divided into Rule Based (Decision Trees, Production rules), Distance Based (kNN, Centroid) and Neural.

Single-label classification is concerned with learning from a set of documents that are associated with a single label (class) l from a set of labels L. In multi-label classification each document can be associated with more than one label from L. If L contains exactly two labels the learning problem is called binary classification, and if L contains more than two labels the problem is called multi-class.

4.1.1. Implemented Algorithms

Our system has a module for automatic multi-label multi-class categorization of documents. The algorithms currently included in the module are Naïve Bayesian, Relative Entropy and Class featured centroid classifier. The conducted experiments showed that the above algorithms provide reasonable classification accuracy and are much faster than more complex methods (such as Support Vector Machines). Nevertheless, the above list is not final, as we are continuously experimenting with new classification methods and strategies to be included in later versions of the system.

4.1.2. Solving the Classification Task

The classification task involves two base phases – training and classifying. The training phase processes a set of labeled documents to create a model. The classifying step uses the model to assign one or more labels to unlabelled documents.

In order to create a model the module represents each document as a set of features. These features are later used to create models for the

different classes. Depending on the classification algorithm a feature reduction method could be applied during the processing.

An LPC processes each document and provides access to different types of features – tokens, lemmas, noun phrases, head tokens. This allows one algorithm to be set up to work with different types of features. Moreover, the categorization module can host several algorithms simultaneously. The results from the different classifiers are combined and the classification result is determined by a majority voting system.

The diagram in Figure 3 depicts the main steps involved in a document classification task:

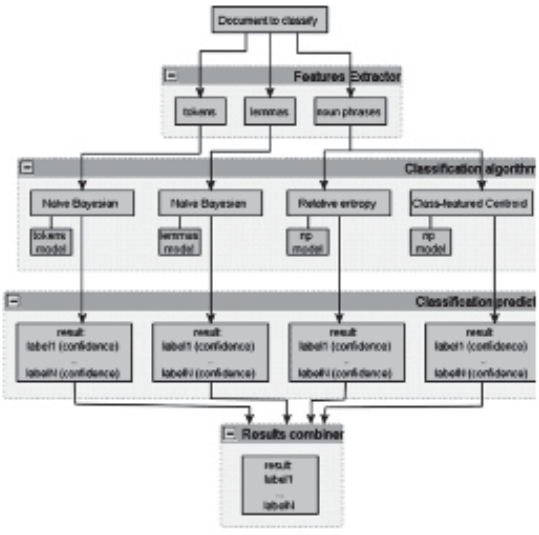

Figure 3: Main stages of the document classification process.

In the example above, there are four classifiers – Naïve Bayesian with tokens feature space, Naïve Bayesian with lemmas feature space, Relative entropy with noun phrases feature space and Class-featured centroid with noun phrases feature space. The classification module registers these algorithms as OSGi services, according to the configuration settings. The features of a new document are extracted by the LPC framework and are passed to the corresponding classifier. Each of the classifiers uses its model to predict a set of labels. Finally, all results are combined and the classification is displayed to the user.

4.2. Machine translation

Machine Translation is a key component of the ATLAS – WebCMS, and it will be embedded in i-Librarian for "translation for assimilation" purposes. The development of the engine is particularly challenging as the translation should be used in different domains and on different text-genres. Additionally most of the language pairs considered belong to the less resourced group, for which bilingual training and test material is available in limited amount.

The machine translation engine is integrated in 2 distinct ways into the ATLAS platform:

In i-Librarian and EuDocLib (see subsection 6.1) the MT-engine provides a translation for assimilation, which means that the user retrieving documents in different languages will use the engine in order to get a clue about the documents, and decide if he will store them. If the translation is considered as acceptable it will be stored into a database.

The integration of a machine translation engine into a web based content management system in general and the ATLAS system in particular, presents from the user point of view two main challenges:

1. The user may retrieve documents from different domains. Domain adaptability is a major issue in machine translation, and in particular in corpus-based methods. Poor lexical coverage and false disambiguation are the main issues when translating documents out of the training domain.
2. The user may retrieve documents from various time periods. As language changes over time, language technology tools

developed for the modern languages do not work, or perform with higher error rate on diachronic documents.

With the currently available technology it is not possible to provide a translation system which is domain and language variation independent and works for a couple of heterogeneous language pairs. Therefore our approach envisages a system of user guidance, so that the availability and the foreseen system-performance is transparent at any time.

Given the fact that the ATLAS platform deals with languages from different language families, and that the engine should support at least several domains an interlingua approach is not suitable. Building transfer systems for all language pairs is also time consuming and does not make the platform easily portable to other languages. Given the user and system requirements corpus based MT-paradigms are the only ones to be considered. In the following we describe the experiments we performed in order to determine the best approach to be used.

Statistical Machine Translation (SMT) is the most used paradigm when the goal of the system is translation for assimilation. The SMT system Moses (Koehn et. al 2007) is not only a translation engine but allows for the development and use of translation and language models by variation of several parameters. We performed several experiments in order to determine if:

- the usual parameter setting used in the evaluation campaigns is suitable also for language pairs in which both languages have a rich morphology,
- the time-consuming tuning step leads to significant improvements,
- PoS-factored models improve significantly the quality of results.

We performed the experiments for all language pairs involving German, Romanian and English, using the parameter setting used in the evaluation campaign from Workshop on Statistical Machine Translation - WMT 2010. As training corpora we used the JRC-Acquis as well as the ROGER-Corpus, a manually aligned domain-specific corpus (Gavrila and Vertan 2011).

Additionally, we compared the results with the Example Based MT (EBMT) system described in (Gavrila 2011). This is a language-independent system, operating at the string level, and embedding linguistic information from the source-input. Following experiments were conducted:

- SMT: comparing the results of our system with results in relevant research papers,
- SMT vs. EBMT on Acquis Communautaire [4],
- SMT vs. EBMT on ROGER,
- ROGER as Test-Corpus for an SMT trained with Acquis Communautaire.
- The experiments lead to following conclusions:
- Even using the same Moses setting, different BLEU (Papieni et. al 2002) scores[5] may be obtained because:
  - test data may be different, and
  - the number of reference translations varies.
- Only one reference translation induces lower BLEU scores;
- BLEU and TER [6] do not always correlate, i.e. BLEU is increasing, but TER is lower or unchanged. This may be an indication that BLEU is looking only at the vocabulary and

4 Acquis Communautaire is the accumulated legislation, legal acts, and court decisions which constitute the body of European Union law. The JRC-Acquis is a collection of parallel texts in 22 languages produced from this resource by European Commission's Joint Research Centre.

5 BLEU (Bilingual Evaluation Understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the correspondence

6 Translation Error Rate is an error metric for machine translation that measures the number of edits required to change a system output into one of the references.

n-grams, while TER mimics „some syntax".

- Even with these disadvantages, running an SMT-system with the classical setting from the evaluation campaign, leads to results similar with those reported in the literature, as shown in Figure 4.
- Tuning is extremely time consuming and improvements are minimal.
- Factored models with PoS improve slightly the evaluation scores.

Regarding the corpus size, which is a very important issue when working with less-resourced languages, our experiments have shown the following:

- Training on a smaller Corpus as Roger (several thousand sentences) does not lead to very bad results, as long as the test data belongs to the same domain.
- The performance is strongly dependent on the accuracy of the word alignment.

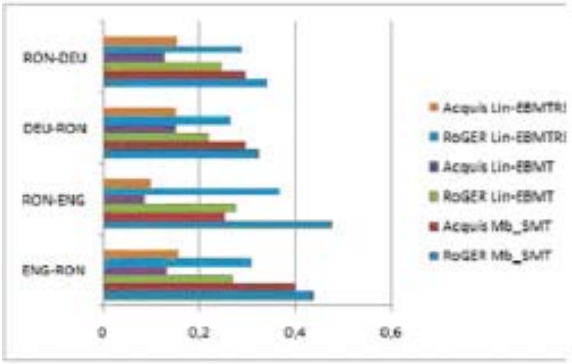

Figure 4. Evaluation for EBMT and SMT Systems

The comparison between SMT and EBMT is summarized below:

- If the sentence or parts of the input are identical with parts in the training corpus EBMT (Somers 1999) is performing better, as the corresponding target language units are automatically retrieved.
- Evaluation scores are lower for EBMT.
- However, after a manual evaluation of approx. 100 sentences, it turned out that the translation quality of several sentences was better in case of EBMT.
- At string level SMT has no possibility to embed linguistic information from the source language which may be relevant for building the translation. EBMT can do this.
- Different domains on training and testing data decrease the performance of the system due to a big number of out-of-domain words.

For the MT engine of the ATLAS system we decided on a hybrid architecture combining EBMT and SMT at word-based level (no syntactic trees will be used). For the SMT component PoS and domain factored models as in (Niehues and Waibel 2010) are used, in order to ensure domain adaptability. An original approach of our system is the interaction of the MT engine with other modules of the system, described below.

The document categorization module assigns to each document one or more domains. For each domain the system administrator has the possibility to store information regarding the availability of a corresponding specific training corpus. If no specific trained model for the respective domain exists, the user is provided with a warning, telling that the translation may be inadequate with respect to the lexical coverage.

The output of the summarization module is processed in such way that ellipses and anaphora are omitted, and lexical material is adapted to the training corpus.

The information extraction module is providing information about metadata of the document including publication age. For documents previous to 1900 we will not provide translation, explaining the user that in absence of a training corpus the translation may be misleading.

The domain and dating restrictions can be changed at any time by the system administrator when an adequate training model is provided. The described architecture is presented in Figure 5.

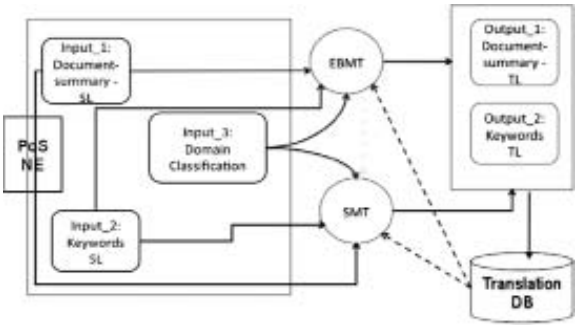

Figure 5. System architecture for ATLAS MT engine

4.3. Text summarization

Summarization of documents in ATLAS applications is intended to present to the interested reader a summary of an article or of a book. The reader could use search facilities to browse the web for an interesting subject and once such a text found, she or he would like to have a quick view over the content of the document that looks interesting at first glance. It could happen that the article is written in a language she or he does not know. A combination of the summarizer and the translation module could present this user with a summary in her/his own language.

There are two types of summaries we are interested in: short texts (short stories or articles up to a few pages) and long texts (for instance, novels). It is clear that in the case of short texts one could indicate the length of the summary as a percentage of the original text, while in the case of long texts this is no more possible, because the variation in length of the obtained summary would be too high. As a consequence, two completely different strategies for obtaining summaries are used in ATLAS: for short texts – based on the identification of the discourse structure, excerpt types of summaries, and for long texts – based on the extraction of relevant information, language generated, template-based, type of summaries. We will describe below only the summarization philosophy used in the project for short texts.

Summarization of short texts in ATLAS benefits from the whole processing chain, while also adding a couple of few other modules to the end of the chain. The initial phases applied to the document being summarized are as follows: identification of paragraphs and limits of sentences, splitting of sentences into clauses, tokenization, POS-tagging and lemmatization, named entity recognition, shallow parsing for the identification of nominal phrases and anaphora resolution. Up to this point, referential expressions (especially pronouns, but also other nominal expressions and name entities) are recognized and they are linked to their antecedents. These coreferential chains help in the identification of the most plausible discourse structure. The discourse structure, a tree, is built incrementally, using a beam-search strategy [7] to limit the exponential explosion of the generated structures. At each moment during the parsing, a wave of N trees (called developing trees), the most promising at that particular moment, is kept, and the rest are dismissed. The following sentence is then parsed and all possible smaller trees (called auxiliary trees) are generated, guided by the discourse markers it contains. Then all these auxiliary trees are combined in all allowed ways with each of the N developing trees, by adjunction on the right frontier (Cristea and Webber, 1997) and substitution (the two tree combination operations inspired by Tree Adjoining Grammar – TAG [8]). With the resulting developing trees, larger than the initial ones with the units of one more sentence, scores are associated based on different heuristics. Then, the whole forest of the resulting developing trees is ranked based on these scores and the best N trees are retained for the next step.

This procedure should end up with a range

7 A *beam search* is a heuristic search algorithm that explores a graph by expanding the most promising node in a limited set.

8 *Tree-adjoining grammars* are somewhat similar to context-free grammars, but the elementary unit of rewriting is the tree rather than the symbol.

of final trees, and, eventually, the best scored is proposed as the discourse structure of the input text. Any percentage-based summary could then be simply trimmed out of the discourse tree. All these summaries have the property of being coherent and pronouns cannot miss their antecedents. Moreover, the veins associated with the discourse units (Cristea, 2009) allow the generation of summaries focalized on certain entities, even if these entities are of minor interest in the text and would not appear in a general summary.

5. Impressions of library users

i-Librarian is currently being evaluated by prospective users. The aim is to assess the acceptance of the online service, by applying indicators that measure the user satisfaction from their experience with the service. The indicators evaluate non functional parameters of i-Librarian, such as:

- user friendliness and satisfaction, clarity in responses and ease of use;
- adequacy and completeness of the provided data and functionality;
- impact on certain user activities and the degree of fulfilment of common tasks.

The primary users of i-Librarian are of three types(i.e. 3 user groups), namely:

1. UG1 – students and scholars: creation of personal library accessible online, formulation of auto-generated multilingual text extracts and document summaries, etc.
2. UG2 – authors, young scientists and researchers: seamless management of various types of documents in different languages, sharing of translated extracts and summaries of papers, articles, etc.
3. UG3 – general Internet users with moderate web experience: creation of personal digital library accessible online, publishing and translation of extracts, etc.

All users are encouraged to try online the service; as an aiding tool a base-line scenario is provided, complemented by an exercise with suggestions about various tasks and steps of activities. The main instrument for collecting user feedback is the interactive electronic questionnaire, available online at http://ue.ATLASproject.eu/ … just select a user group and follow the onscreen suggestions!

Figure 6. i-Librarian interface

6. Other document libraries

Apart from i-Librarian, two other document libraries have been prepared with ATLAS technical and linguistic framework – EUDocLib and PLDocLib, this time mostly for demonstration purposes, with editing actions made unavailable to the general public. They both offer linguistically-aware search.

6.1. EUDocLib

The EUDocLib service [9] is a publicly accessible repository of EU documents from the EUR-LEX collection which provides easier access to relevant documents in the user's language, providing:

- automatically categorized, summarized and annotated content with important noun phrases and named entities,
- better content navigation (such as list of similar documents) based on interlinked text annotations,
- machine-translated excerpts of documents and using them for document categorization and clustering.

Currently the site covers 140 K documents (182 M tokens).

Figure 7. EUDocLib interface: search results

6.2. PLDocLib

The Polish variant of EUDocLib [10] is a language processing chain-powered Web site offering search and browsing of around 1000 legal acts of Polish Parliament (Sejm) automatically annotated with a set of ATLAS-integrated tools for Polish:

- Morfeusz – a morphological analyser for Polish,
- Pantera – a rule-based Brill tagger of Polish,
- Spejd – an engine for shallow parsing using cascade grammars,
- plNER tool – a statistical CRF [11]-based named entity recognizer.

On the basis of the annotations, the Web application provides for each document a set of recognized named entities, important noun phrases (in clusters, based on their similarity and importance) and a list of similar documents. For presentation, base forms of multiword units are generated and manually assigned categories are used.

Figure 8. PLDocLib interface: search results and a sample document with topics and similar documents

9 http://eudoclib.ATLASproject.eu/
10 http://www.ATLASproject.eu.pl/

11 CRF – *Conditional Random Fields* are a class of statistical modelling method often applied in pattern recognition and machine learning, where they are used for structured prediction.

Conclusions

The abundance of knowledge allows us to widen the application of NLP tools, developed in a research environment. The combination of web content management and state of the art language technologies helps the reader to cross the language barrier, to spot the most relevant information in large data collections and to keep all this information in order. The tailor-made voting system maximizes the use of different categorization algorithms. Two distinctive approaches summarize short and long texts and their translation are provided by state-of-the-art hybrid machine translation system.

ATLAS linguistic framework will be released as open-source software. The language processing chains for Bulgarian, Greek, Romanian, Polish and German were fully implemented by early 2012.

We expect this platform to serve as a basis for future development of deep analysis tools capable of generation abstractive summaries and training models for decision making systems.

References

Cristea Dan and Bonnie Lynn Webber. 1997. Expectations in Incremental Discourse Processing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid.

Cristea Dan. 2009. Motivations and implications of veins theory: a discussion of discourse cohesion. In *International Journal of Speech Technology*, 12(2/3), 83–94.

Gavrila Monica. 2011. Constrained recombination in an example-based machine translation system. *EAMT-2011: the 15th Annual Conference of the European Association for Machine Translation*, Eds. Vincent Vondeghinste, Mikel L. Forcada, and Heidi Depraetere, 193–200, Leuven, Belgium: EAMT.

Gavrila Monica and Cristina Vertan. 2011. Training data in statistical machine translation – the more, the better? In *Proceedings of the RANLP-2011 Conference*, Hissar, Bulgaria.

Hohpe Gregor and Bobby Woolf. 2003. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Professional.

Koehn Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbst, 2007. *Moses: Open Source Toolkit for Statistical Machine Translation,* Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic, June 2007.

Niehues Jan and Alex Waibel, 2010. Domain Adaptation in Statistical Machine Translation using Factored Translation Models, *Proceedings of EAMT 2010*, Saint-Raphael.

Ogrodniczuk Maciej and Diman Karagiozov. 2011. ATLAS – The Multilingual Language Processing Platform. *Procesamiento del Lenguaje Natural*, vol. 47, 241–248.

Papineni Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, 311–318.

Somers Harold. 1999. Review Article: Example-based Machine Translation. *Machine Translation,* 14(2):113-157.