DELIVERABLE

Project Acronym:	**ATLAS**
Grant Agreement number:	**250467**
Project Title:	**Applied Technology for Language-Aided CMS**

Deliverable D7.3 Final Report on Test Results

Authors:

Polivios Raxis, D. Botsis, B.Dikeoulias, P.Gravaris, G.Kalamakides.… (Atlantis)
Anelia Belogay, D. Karagiozov ..... (Tetracom)
Dan Cristea, E.Ignat, D. Anechitei .....(UAIC)
Cristina Vertan, R.Winnemöller.....(UoH)
Maciej Ogrodniczuk, Adam Przepiórkowski.....(ICS PAS)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	**X**
C	Confidential, only for members of the consortium and the Commission Services	

Revision	Date	Author	Organisation	Description
0.1	01/11/2012	P. Raxis	ATLANTIS	First complete outline of the document
0.2	27/12/2012	P. Raxis, D. Botsis, B.Dikeoulias, P.Gravaris, G.Kalamakides	ATLANTIS	Introduction, methodology, assessment of specs fulfillment, integration/regression testing
0.3	21/01/2013	D.Cristea, E.Ignat, D. Anechitei	UAIC	Technical evaluation of summarization
0.4	30/01/2013	Anelia Belogay, D. Karagiozov	TETRACOM	Technical evaluation of categorization
0.5	31/02/2013	D. Botsis, B.Dikeoulias, P.Gravaris, G.Kalamakides	ATLANTIS	Technical evaluation of CMS and LPCs
0.6	01/02/2013	D. Karagiozov	TETRACOM	Integration and regression testing: implementation and results
0.7	04/02/2013	C.Vertan, R.Winnemöller	UoH	Technical evaluation of MT and CLIR
0.8	06/02/2013	Maciej Ogrodniczuk, Adam Przepiórkowski	ICS PAS	Revision of Polish LPC data and corrections in the whole deliverable
0.9	06/02/2013	A.Belogay, D.Karagiozov, E.Stoyanov	TETRACOM	Revision of final version

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This deliverable contains original unpublished work except where clearly indicated otherwise.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.

CONTENTS

1 INTRODUCTION

1.1 *Overview*

D7.3 is the third deliverable of the WP7 "Testing and User Evaluation" of the ATLAS project.

The ATLAS project aims to unify and integrate mechanisms for automatic annotation of important words, phrases and names, text summarization and categorization and computer-aided translation in a process of manipulating heterogeneous multilingual content in a common software platform and as a result to deliver three software-as-a-service solutions, which offer all the tools individuals and organizations need to manage their multilingual content.

The first solution, **i-Publisher**, adds a visualization layer to ATLAS and provides a powerful web-based instrument for creating, running and managing small and enterprise content-driven web sites. The second solution, **i-Librarian,** allows its users to store, organize and publish their personal works, to locate similar documents in different languages, and to easily obtain the most essential texts from large collections of unfamiliar documents.

These two solutions are empowered through the main ATLAS developed components, namely:

1. LPC: provides annotations (tokens, PoS, lemma, named entities, etc.) on input documents in all project languages.
2. Categorization: creates a categorization model for the provided parameters and categorizes automatically previously unseen text content using appropriate models.
3. Summarization: provides an automatically generated summary of an input text.
4. Machine Translation: utilizes two engines - example-based MT and statistical MT. The results of both engines are blended in order to provide a translated version of an input text.
5. Cross-lingual IR: uses the translated data from MT and performs cross-lingual information retrieval.

The deliverable D7.3 "Final Report on Test Results" contains details about:

- The testing scope, with respect to what is to be tested, what is the scope with respect to individual components and indicators, etc.
- The overall methodology to be followed: how the activities will be organised and conducted, what will be the methodological approach to the different testing and evaluation challenges, processing of results, etc.
- The methodology and the approach used for testing the ATLAS platform, at the level of isolated components, at the level of integrated platform, and after each new deployment (i.e. regression testing).
- The methodology used for assessing the level of ATLAS application specifications fulfilment, including the assessment results for its two main applications (i-Librarian, i-Publisher).
- The collection of documents (corpora) used for both testing and technical evaluation.
- The methodology used for the technical evaluation of the platform components, including test cases and scenarios for each main component and respective technical indicators.

- The findings from the technical evaluation of each main component, including comparative assessment results – where this was applicable.

The document is organised into 5 main chapters:
Chapter 1, provides an overview of the "object to be tested" and presents the scope and objectives of the report.

Chapter 2, describes the methodological aspects of the testing and evaluation, including the methodology to be followed for integration and regression testing, the methodology and the indicators for the technical evaluation, etc.

Chapter 3, focuses on the integration and regression testing of the whole platform and its main components, outlining scenarios, testing steps and conditions for failure and success, testing corpora, testing results, etc.

Chapter 4, presents the results of the assessment of platform and applications' specifications fulfillment, including the defined simple-confirm indicators and the respective testing results.

Chapter 5, details the results of the technical evaluation of the main ATLAS components (machine translation, categorization, summarization, etc.), including test cases, respective technical indicators, etc.

1.2 *Used abbreviations*

CMS	Content management system
MT	Machine translation
SMT	Moses-based Statistical Translation
EBMT	Example Based Machine translation
LPC	Language Processing Chain
POS tagger	Part of Speech tagger
NP extractor	Noun phrase extractor
CLIR	Cross-lingual information retrieval
(P)	Precision
(R)	Recall
FM	F-measure
NER	Named Entity Recognizer

2 METHODOLOGY

2.1 *Overview of the approach*

The ATLAS software platform integrates and harmonizes numerous heterogeneous tools and 3rd party libraries. As a network distributed application, ATLAS software and its components use various communication patterns - from the classical request/response blocking pattern to the asynchronous messaging patterns. Furthermore, the Atlas components work in a sequence. As an example, the automatic summarisation and categorization components use the output of the Language Processing Chains (LPC) module in order to execute their core methods.

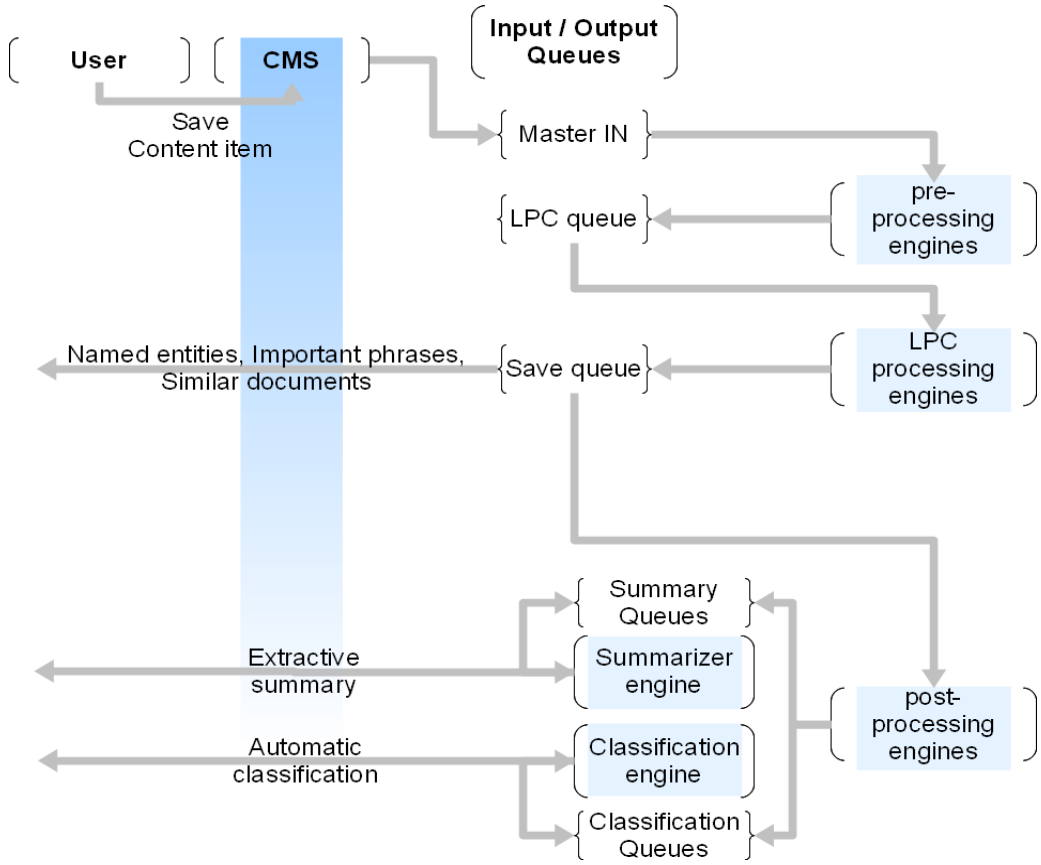
The testing of the ATLAS platform and its components involves verification of correct execution of the individual platform constituents and of the 2 applications build on the top of the platform, integration tests across the ATLAS integrated s/w platform, regression tests for every new system build produced, assessment of system specifications fulfilment and technical evaluation of the main platform components. To this end, our approach includes different methodology for:

- *Isolation testing*: s/w testing of the various ATLAS components and modules (LPCs, summarisation, message queue, communication interfaces, etc.), and the 2 applications (i-Librarian and i-Publisher). Primarily based on defined JUnit tests integrated in the respective s/w fragments. In addition, application testing scenarios were executed for each application.
- *Integration and regression testing*: definition and development of an automated s/w integration testing and a semi-automated regression testing infrastructures. These are deployed in order to detect integration failures when components and modules are exchanging data; and to detect problems introduced in the system from new deployments.
- *Specs fulfilment*: definition of simple-confirm indicators to assess the level of fulfilment of platform specifications. Each main specification is mapped to such an indicator and assessment is made as of the degree of the implementation of the respective ATLAS functionality, within the two main applications.
- Technical evaluation: definition of technical indicators for each ATLAS main component, appropriate for the technical characteristics of the component under evaluation (CMS and LPCs, summarisation, MT, etc.). The indicators refer to efficiency and detection rates, to precision, recall and F-measure, to BLEU scores, etc. and are meant to assess the technical performance of the ATLAS main components.

The subsections following provide details for each methodology.

2.2 *Isolation, Integration and Regression testing plan*

In order to ensure the smooth work of the ATLAS platform, we developed an automated s/w testing and semi-automated regression testing infrastructure. Such infrastructures support the identification of problems and deviations in the quality caused by the deployment of new versions of components, changes and bug fixes, communication failures, etc.



The following initiator, mediator and functional components are identified:

- The user - the users interact with ATLAS and its services through a web browser. ATLAS frontend is based on ZK RIA libraries which support all major browsers and their versions. The scope of the deliverable **does not** cover any tests performed on the client side.
- i-Publisher and i-Librarian CMS - this component provides all content management functionalities and integrates the functionalities provided by the rest of the ATLAS components.
- Pre-processing engine - the pre-processing engine is responsible for detecting the mime type of input documents, for extracting the text from recognized document sources and for detecting the language of the extracted text.
- Language processing chains - the LPC engines, at least one per language, enrich the provided text with linguistic annotations.
- Post-processing engine - the post-processing process stores the linguistic annotations in a hybrid datastore (a fusion between RDBMS and Lucene indexes).
- Categorization engine - has two responsibilities; i) to create a categorization model for the provided parameters and ii) to categorize automatically previously unseen text content using appropriate models.
- Summarization engine - the summarization engine provides a reduced version (summary) of the input text.

- Translation infrastructure - the machine translation engine in ATLAS utilizes two engines - example-based MT and statistical MT. The results of both engines are blended and shown to the user.
- Cross-lingual IR engine - the CLIR engine uses the translated data from MT and performs cross-lingual information retrieval.
- Relational database - most of the data managed through ATLAS resides in a PostgreSQL relational database.
- Lucene indexes - text annotations data is stored in Lucene indexes in order to increase the overall system performance and responsiveness.
- Message queues - ApacheMQ is used as major communication infrastructure between the ATLAS components. Our tests focus on the communication between ATLAS components and the message broker.

As a starting point all the above individual modules (pre-processing, post-processing engines, queues and communication with the message broker, etc.) were tested in isolation using Unit Tests. Furthermore, individual LPC modules (tokenizer, NP extractor, lemmatizer, NER, etc.) were tested and verified for each project language during the course of WP4 activities; the approach and the results are documented in the D41 deliverable. Similarly for the main ATLAS components, namely Categorisation, MT, CLIR, and Summarisation (WPs 3, 5, 6 and respective deliverables).

In particular for the two main ATLAS applications, i-Librarian and i-Publisher, we have defined JUnit tests (integrated in the application code) which are executed when a new version of the applications is deployed. Failures are recorded in the respective application logs and addressed by the application development team.

Apart from testing components and applications in isolation (i.e. testing and verification at component level), we produced scenarios and sequences of steps in order to test the integrity and efficiency of the integration of all components and modules under the integrated ATLAS software platform. In doing this, we collected a test corpora with documents from all project languages; and we fully annotated each of these documents in order to have a ground-truth for our automated tests.

Apart from the integration tests, we use this test corpus also for regression tests. Whenever we produce a new ATLAS system build, we execute a sequence of testing steps, part of our regression test scenario. In this way, whenever a failure is detected in any step (i.e. by comparing step output with the last good known output of this step), we keep track of the problem, we apply corrective measures, and we re-iterate the testing steps until no problems or failures are detected.

In chapter 3, “Integration and Regression testing” we focus on the details of the integration and regression tests and the corpora used. Details of the isolation tests (components, modules, applications, etc.) are not provided in this document, to avoid overloading of the reader with too many low level details.

2.3 *Assessment of Specs Fulfilment*

It is our intention to assess the level of fulfilment of platform specifications, as these were recorded in the document specified the functional requirements of the main 2 ATLAS applications; i-Librarian and i-Publisher. In order to do this we defined simple-confirm indicators; each main specification is mapped to such an indicator and assessment is made as of the degree of the implementation of the respective ATLAS functionality, within the two main applications. The indicators test the existence or absence of the particular functionality and record findings in a tabular form, as following:

Simple Confirm Indicators	Success / YES	Failure / NO	Remarks
Indicator-1: related to functional spec-1	☐	☐	
Indicator-2: related to functional spec-2	☐	☐	
…….	☐	☐	

If a particular functionality is available, then the respective box is checked. Otherwise, NO is checked, and a remark is noted.

2.4 *Technical Evaluation*

Our objective was to assess the technical performance of the ATLAS platform by evaluating technically the performance of its main components (summariser, categorisation, MT, etc.). For this reason we defined technical indicators for each main component, and we collected measurements for each indicator in order to draw justifiable conclusions. The indicators refer to efficiency and detection rates, to precision, recall and F-measure, to BLEU scores, etc.

The measurements were collected based on one or more Test Cases which provided the methodology for the technical evaluation approach to be used for each component. Where needed, dedicated test corpora were produced to provide the necessary data sets for the implementation of the technical evaluation experiments. O the completion of each experiment, we analysed the findings and rectified malfunctions, addressed weaknesses, etc.

Chapter 5, "Technical Evaluation of Platform Components" provide details on the performed technical evaluation of the ATLAS components. Any components or modules developed outside the ATLAS framework are not included in the scope of our technical evaluation.

3 INTEGRATION AND REGRESSION TESTING

3.1 *Introduction*

As presented in the previous chapter (Methodology), we developed an automated s/w testing and semi-automated regression testing infrastructure. These consist of scenarios and sequences of steps in order to i) test the integrity and efficiency of the integration of all components and modules under the integrated ATLAS software platform; ii) detect in any step (i.e. by comparing step output with the last good known output of this step) problems introduced into the platform from the deployment of new build. These scenarios and sequence of steps make reference to individual steps, which are outlined below:

i-Publisher and i-Librarian tests

i-Publisher and i-Librarian software application components are equipped with JUnit tests which are executed for every test and productive deployment. The unit tests cover the major CMS workflows and main functionalities. Apart from these tests which are executed automatically, we execute manually (and record problems in order to be addressed) the following application scenarios:

1. User Test Scenario 1 (i-Librarian) - http://ue.atlasproject.eu/uts1
2. User Test Scenario 2 (i-Publisher) - Choose from the Atlas websites and fill out content - http://ue.atlasproject.eu/uts2
3. User Test Scenario 3 (i-Publisher) - Customise a theme and build a website - http://ue.atlasproject.eu/uts3
4. User test scenario 4 (i-Publisher Advanced mode) - http://ue.atlasproject.eu/uts4

Pre-processing tests

The key functionalities of the pre-processing engine are tested in a chain:

`mime-type detection → text extraction → language recognition`

The tests run automatically on all documents in the test corpora. The tests compare the manually provided set mime type, the size of the textual content and the language with the features provided by the pre-processing engine.

LPC tests

The LPC testing focuses on the input and output of every subsequent module in order to identify and avoid accumulation of errors in the chain. We have manually annotated a subset of the documents in the test corpus (see section 3.2 "Testing corpora") with sentence, token, PoS, lemma, noun phrase and named entity annotations. The automatic test compares the manual and the LPC-provided annotations for each document in the annotated corpora. This test fails if the precision is lower than a predefined threshold. The thresholds are different for each LPC and for each individual linguistic tool (i.e. primitive engine), and within the context of each tool, for each document in the respective manually annotated corpus. For each threshold, its value is determined by comparing the selected manually annotated document against the annotated file automatically produced by the current system deployment. E.g. for a given GR manually annotated document, auto-annotation of the last good deployment identifies correctly 129 POS tags for 140 tokens existing in the document; this will be considered as the threshold value.

In addition, the `uimaFIT` framework is extended to accommodate the specific needs of ATLAS and is used as a base for the unit tests. In that way all functional LPC levels (components, UIMA integration and chain execution) are fully covered by an automatic testing.

Post-processing tests
The data integrity is the key aspect tested for the post-processing engine because of the usage of hybrid data. The automatic tests compare the number of valid input annotations (sentences, tokens, named entities and noun phrases) with the number of the stored annotations in the database and in the Lucene indexes. The test may fail in two cases:

1. The number of stored and valid input annotations differ;
2. There are invalid annotations. In this case all invalid annotations are recorded for further handling and improvements.

Categorisation tests
The quality of the results of the categorization engine strongly depends on the quality of the training and test data. In order to assess the quality of the categorization algorithms, the tests are performed on the well-known and scientifically recognized Reuters-21578 corpus. We have initially recorded precision, recall and f1-measure for all 90 categories, as well as the micro- and macro-f1-measure for the whole data set. The automatic tests compare these recorded values with the results from the application of a newly built model to the document in the data set. A test fails if the difference between recorded and experienced f1-measure is greater than a predefined threshold.

For each system category (80 top-level categories are being used in i-Librarian), we test the multilingual aspects of the automatic categorization. The automatic test builds the categorization models for each language and checks its validity by categorizing the training documents. This test fails if the f1-measure for each category is not in a predefined interval. The upper boundary of this predefined interval is set to 0.93 in order to avoid model over-fitting.

Summarisation tests
The summarisation component is made up of a chain of 4 processes, each being fed with the output of the LPC chain. The 4 processes go in sequence as follows: anaphora resolver (RARE – a module finding referents for anaphors, mainly pronouns, but also other reference expressions), segmenter (for segmenting sentences down to clauses, considered elementary discourse units), discourse parser (building discourse trees out of the elementary discourse units and using for that information provided by RARE), and the summariser itself (which is able to extract the most important discourse units out of the structures built by the discourse parser). In order to avoid accumulation of errors in the chain, the 4 above modules are tested independently, by comparing their output on a test file (1 per language) against a gold file.

Machine translation tests
The machine translation component consists of two units: the example based engine (EBMT) and the Moses-based statistical engine (SMT). Both of them rely on a parallel corpus aligned at word level. While the SMT uses a statistical model computed from the parallel corpus, the EBMT uses the parallel corpus as a translation database.

The ATLAS MT engine uses first the EBMT module and checks if the input sentence can be found (entirely or a bigger part of it) directly in the parallel corpus. In this case the EBMT module produces the output. In all other cases the SMT engine is called. Our test ensures that input, identical to parts of the parallel corpus, is identified as such and therefore the EBMT engine is started. For this, we have extracted a set of 50 sentences and their English translations from the JRC-Acquis in all project languages. These input sentences are passed through to the EBMT engine. This automatic test is successful, if the results are identical to the original translations.

The statistical MT engine should be used whenever the input is not close enough to the parts of the parallel corpus. To test that the communication works properly, we have isolated 50 sentences from the corpora used for the translation models in SMT (they are not be included in the translation memory of the EBMT). Together with the 50 sentences used in EBMT, these form a test set of 100 sentences for the MT engine. The test is successful if it leads to a mixed output from EBMT and SMT engine. The test fails, if the MT component picks translations from the wrong MT engine.

Cross-lingual IR tests
We have selected documents in all project languages other than English for 10 topics in Wikipedia; 10 queries are formulated in English. The test is successful if the CLIR returns all documents.

Messaging tests
The communication channels between the major components are based on synchronous and asynchronous message exchange. The automatic test sends messages to specific queues and expects acknowledgement that the message has been accepted. The following messages are tested:

- i-Publisher → pre-processor - message contains binary data;
- pre-processor → specific LPC - message contains text;
- specific LPC → post-processing - message contains annotated text (XML);
- post-processing → automatic categorization - message contains document ID;
- post-processing → summarization - message contain XML;
- i-Publisher → automatic categorization - message contains document ID;
- i-Publisher → MT engines - message contains source and target languages, as well as the text to be translated;
- i-Publisher → CLIR engine - message contains the text of a query.

The test fails if the acknowledgement has not been received within a preset period of time.

3.2 *Testing corpora*

The test data have been gathered by the ATLAS partners to be used internally in integration and regression tests. The sets (one for each project language) intend to include documents representing different formats and most popular encodings used for particular language as well as different sizes (with at least one document exceeding 100K tokens). The description of data used for internal tests is available online at http://www.atlasproject.eu/wp4/test-data-internal.pdf.

We provide the following additional features for each of the document in the initial test corpus:

- mime type
- size of the textual content
- language of the document

Manually annotated documents

The table below lists the documents that are manually annotated and used as reference results for the automated regression tests (i.e. comparisons between ATLAS auto-produced annotations on these files, and the manually annotated files). The first column is the language and the second one contains the IDs of the test documents for this language.

BG	BG-01, BG-06, BG-08, BG-17, BG-19, BG-23, BG-28, BG-39
DE	DE-04, DE-07, DE-08, DE-11, DE-13, DE-19, DE-21, DE-23, DE-24, DE-30
EL	GR-01, GR-04, GR-100, GR-10, GR-18, GR-20, GR-50
EN	a1-a5 (5 articles), n1-n9 (9 news stories)
PL	PL-06, PL-07, PL-08, PL-10, PL-12
RO	RO-20120522-* - (these are 19 new test documents, all manually annotated)

Reuters-21578

Reuters-21578 collection Apte' split includes 12,902 documents for 90 classes, with a fixed splitting between test and training data (3,299 vs. 9,603). The categories are represented as different directories. The files (one for each document) associated with the target category are stored in the corresponding directory. This collection is mainly used for testing the Categorisation module.

3.3 *Integration Testing*

The tests outlined in the previous subsection refer to automated tests of individual components participating in the integrated ATLAS platform. Below we outline a testing scenario for the integrated workflow, using the i-Publisher as a starting / reference point for the various steps involved:

1. one test document for each language is sent to the pre-processor;
2. the scenario engine waits for acknowledgement message;
3. upon such a message, the scenario engine requests data set, containing:
   - the extracted named entities - persons, organizations and locations;
   - the most important phrases;
   - the assigned categories;
   - the produced summary;
   - the translations of the:
     - ✓ most important phrases
     - ✓ produced summary

4. retrieved data is compared with the recorded data.

The scenario fails if any of the above steps produces a result (e.g. message, data set) worse than the expected (e.g. respective result recorded from the last good known system integration / deployment). Furthermore, intermediate produced annotated files are compared against the manually annotated files.

3.4 *Regression Testing*

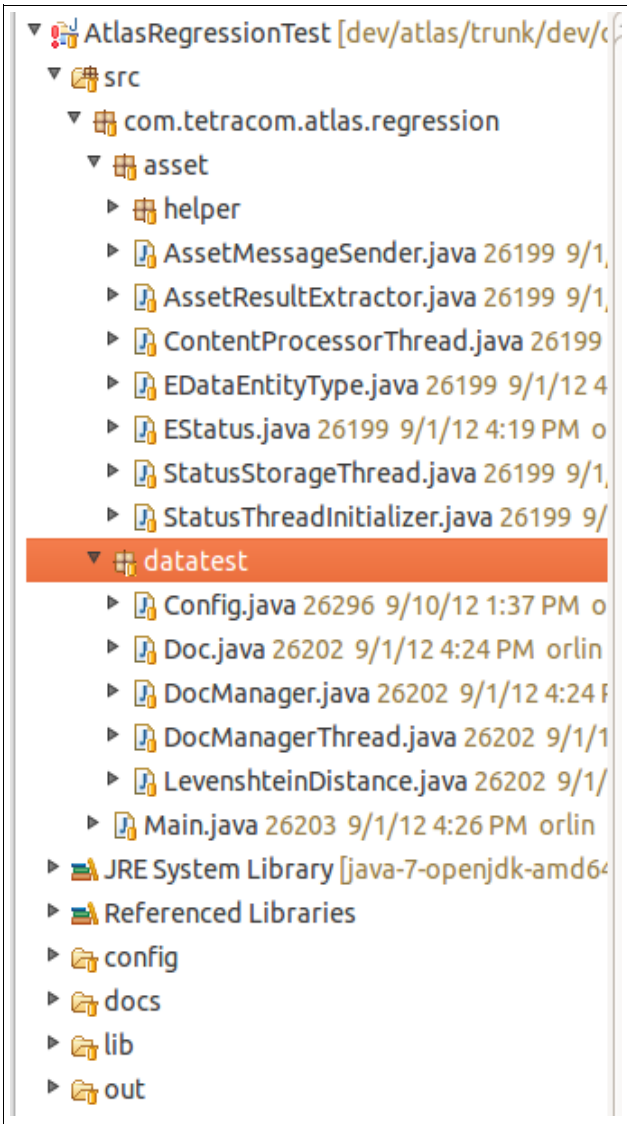
When a system upgrade is committed to the project repository and a new system build is produced, before deploying this build to the production environment, an overall regression test is executed automatically. In essence, this test incorporates the individual component (and integration flow) tests described in previous subsections:

- i-Publisher, i-Librarian
- Pre-processing
- LPC
- Post-processing
- Categorization
- Summarization
- MT and CL IR
- Messaging
- Integration flow

The overall regression test is facilitated by an automated executable scripting process which orchestrates all individual tests. Whenever the regression test is executed, execution logs are generated automatically and compared to the last good known deployment in order to identify newly introduced problems. Furthermore, intermediate produced annotated files are also compared against the manually annotated files.

3.5 *Test implementation*

Two approaches have been used in the implementation of the regression tests for the components in ATLAS. iPublisher was initially designed with JUnit integration in mind, thus the regression tests for this service are based on the JUnit toolkit. The regression tests for all other components are based on a custom infrastructure. This choice is dictated by the complexity, prerequisites and dependencies among the tested components. For example, categorization and summarization tools depend on the LPC output, the machine translation and CLIR engines depend on the output of the summarization tool. Our regression tests implementation is packaged as a Java ARchive (JAR) and is invoked from the command line (shell).



The source core is organized as follows:

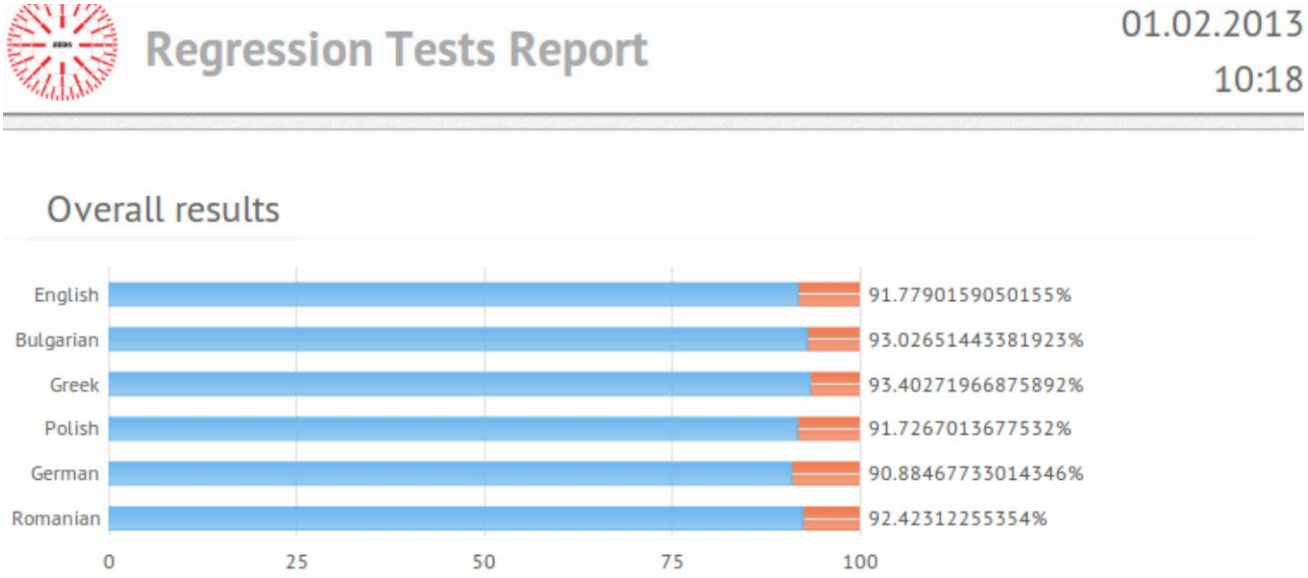
- the tool's configuration is located in the “configs” folder. There we describe: (a) the kind of the tests that should be performed; (b) where are the last known-to-be-good results; (c) the dependencies between the tests (e.g. process documents via LPC, then store, then categorize, finally summarize them); (d) how long to wait for results; etc.
- “docs” folder contains the last known-to-be-good results for each language. Each document and its annotations are kept as a pair of files: (a) one contains the full text of the document; (b) the other the expected annotations.
- The classes in the “com.tetracom.atlas.regrassion.asset” package implement the communication between the regression tool and ATLAS.
- The classes in the “com.tetracom.atlas.regrassion.datatest” package compare the results from ATLAS with the last known-to-be-good results. A class here produces an HTML report which summarizes the results of the regression tests.
- The Main.java class starts and orchestrates the regression tests based on the configuration.

3.6 *Testing Results*

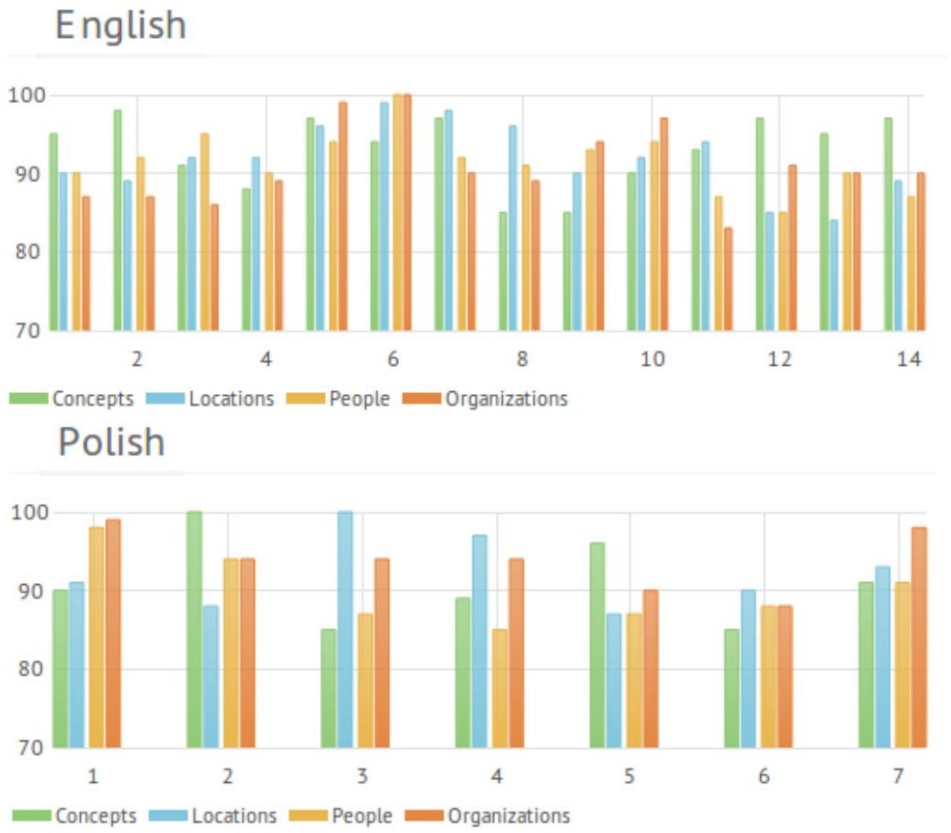
The results of each regression test are presented in the form of a report. For some of the ATLAS components (e.g. Machine Translation, Summarization and CLIR engines) we cannot formally measure the quality of the returned results (translation, summary or multilingual search hits), thus the report contains only information whether the component is working and if there were results returned. For the LPCs, however, a more detailed report is generated.

Firstly, the reports give an overview of the quality of the LPCs for each language. The quality is represented as a score in the range 0 – 100. “0” means that the corresponding LPC failed in all tests, e.g. no results have been returned in half an hour; “100” means that all results returned by the LPC match the previously known-to-be-good results. Seldom a LPC has a quality of 100 because the returned results (named entities and noun phrases) are sorted by a modified version of the standard tf*idf weight. As the inverted document frequency (idf) changes when new documents are added to a collection, the weight of each named entity or noun phrase differs with each run of the regression test. A score above 85 is accepted as pass, scores between 75

and 85 are acceptable, scores less than 75 is an indicator of a potential problem which needs further investigation. The screenshot below depicts the test results overview for the 6 languages in ATLAS.



The bars in the charts below show the quality score for each document for each of the evaluated entities (noun phrase/concepts, locations, people and organizations). The next screenshot shows the detailed results of the regression tests for English and Polish languages. Currently, 14 documents are compared for English, 8 for Bulgarian, 7 for Greek and Polish, 10 for German and 19 for Romanian.



4 ASSESSMENT OF SPECS FULFILLMENT

4.1 *Introduction*

One method of technical evaluation is the use of “Simple Confirm Indicators”. In this method, the existence of a specified (i.e. as documented in the specification of the ATLAS functional requirements) ATLAS functionality is verified through a simple test (e.g. by examining the system functionality and confirming the existence or absence of the particular functionality). The tables following present the results of simple confirmation of the two ATLAS main services: i-Publisher, i-Librarian.

If a particular functionality is available, then the respective box is checked. Otherwise, NO is checked, and a remark is noted.

4.2 *Simple confirm testing results*

i-Publisher Simple-Confirm Indicators

Simple Confirm Indicators	Success / YES	Failure / NO	Remarks
Means available to the users to register and logging in	☒	☐	
A registered regular user may log-in, view and edit his profile and retrieve a forgotten password	☒	☐	
An anonymous user can view websites created with the system unless a website has been restricted to specific users by its owner	☒	☐	
The system stores info (contact, account, domain) for each registered user	☒	☐	
A super-user (i.e. admin) can manage (create, change, activate, delete, set access rights) user and user-group accounts	☒	☐	
A regular user can perform in their own domain the same actions as a super user, provided that the user has sufficient privileges	☒	☐	
The system provides a built-in core domain, which is accessible only by the super user. It includes data model, core vocabularies and taxonomies	☒	☐	
The super user can modify the entities in the core domain, export/import entities, and manage (create, modify, disable/enable, delete, make websites online/offline) all domains	☒	☐	
A regular user (if they have sufficient privileges, and only in their own domain) can modify the entities in his domain, export/import entities, and manage (modify, disable/enable) his domain	☒	☐	
The system provides a web-based interface for content authoring and management	☒	☐	
A regular user can (if they have sufficient privileges) manage the content model in their domain, the approval workflows, the controlled vocabularies and taxonomies	☒	☐	
A regular user can (if they have sufficient	☒	☐	

Simple Confirm Indicators	Success / YES	Failure / NO	Remarks
privileges) manage (create, modify, delete, classify, preview) content in his domain			
A regular user can (if they have sufficient privileges) view the history of a content item, create a content item revision and restore a content item to a previous revision	☒	☐	This functionality is accessible from Advanced Mode, Content item details. User should explicitly select “Create content version” link in order to create new version of a content item. There is a link in the menu called “Content versions” which displays all previous versions of a content item. In order to restore previous version of a content item, the user selects a version from the list and clicks on Restore button.
A regular user can (if they have sufficient privileges) manage (create, discard, modify, populate, view and filter) selections of content items	☒	☐	
A regular user can (if they have sufficient privileges) manage collections (create, modify, remove, add, view)	☒	☐	
The system provides a web-based interface for website building and management	☒	☐	
A user can (if they have sufficient privileges) list their websites in their domain and import / export a redistributable website package	☒	☐	
A user can (if they have sufficient privileges) create, modify and delete a multilingual website	☒	☐	
A user can (if they have sufficient privileges) preview, publish and make a website offline (e.g. for maintenance)	☒	☐	
A user can (if they have sufficient privileges) create a website snapshot and revert a website to a previous snapshot	☐	☒	At the moment a web site can be exported as a package but it includes only structure (pages, widgets, text and images, content types) and does not include content items. Afterwards this package can be imported as a new web site which can replace the original in case this is needed.
Once a website has been published, the system makes it available at the address chosen by the website owner	☒	☐	
A website visitor can switch between the different website languages	☒	☐	
The system makes each page in a website available at the address and alias addresses specified by the website owner	☒	☐	
The system allows users to filter content using any combination of context filters (classification, text mining, and search filters)	☒	☐	This functionality is implemented in “Content item list” widget dialog, where the user can implement different ways of sorting, grouping, filtering and

Simple Confirm Indicators	Success / YES	Failure / NO	Remarks
			paging. Also, the user can specify a data filter for the widget, which provides specific logic based on the content item's properties.
Context filters can be used by all users (with sufficient privileges) as well as by website visitors, provided that the website owner has activated the necessary optional website functionality	☒	☐	
The system provides a web-based interface that allows users (with sufficient privileges) to create new or adjust existing classification models	☒	☐	This functionality is present in i-Publisher Advanced mode and it is accessible via "Core domain area" perspective , "Add new categorization tree" link from the menu on the left. Within the interface there is an "Automatic categorization" menu item, where user can build and reset models for this particular tree and for each available language in the domain. There is also Model configuration and Training data sections where user can perform the described functionality.
The system provides a web-based interface that allows users (with sufficient privileges) to specify the classification method and model to be used, as well as the text properties of a content type that will be used in automatic classification of content items of that type	☒	☐	
The system provides a web-based interface that allows users (with sufficient privileges) to specify the text properties of a content type to be used to create summaries for content items of that type	☒	☐	This functionality is implemented on a higher level and is done via setting a content item property to be used for any kind of text analysis, including summarization.
The system provides a web-based interface that allows users (with sufficient privileges) to specify the summarization method to be used	☐	☒	This functionality is yet to be enabled, since at the moment summarization is still experimental.
The system provides a web-based interface that allows users (with sufficient privileges) to specify the translation model to be used	☐	☒	This functionality is yet to be enabled, since at the moment machine translation is still experimental.
The system provides a web-based interface that allows users (with sufficient privileges) to specify the text properties of content items of a given content type to be translated and into what languages	☒	☐	
The system can automatically extract from content items: noun phrases, named entities and References (URLs and e-mail addresses)	☒	☐	

i-Librarian Simple-Confirm Indicators

Simple Confirm Indicators	Success / YES	Failure / NO	Remarks
An anonymous user (who has not registered or not logged	☒	☐	

Simple Confirm Indicators	Success / YES	Failure / NO	Remarks
into i-Librarian) can view public content items (documents), or register			
A regular user can Log in/out, view/edit his profile, retrieve a forgotten password	☒	☐	
Regular users can in their own workspace set the access rights to specific content and operations with those content items	☒	☐	
The system stores info (contact, account) for each registered user	☒	☐	
A super-user (i.e. admin) can manage (create, change, activate, delete, set access rights) user accounts	☒	☐	
Regular users can, in their own workspace (when logged in) upload a content item in various formats – PDF, DOC, XLS, PPT, LIT, TXT, RTF	☒	☐	
Regular users can, in their own workspace (when logged in) choose what meta-data formats are associated with the uploaded content item	☒	☐	
Regular users can, in their own workspace (when logged in) export content items and import content items shared by other users, or through mass-imports (e.g. from ZIP, RAR archives)	☒	☐	
Regular users can, in their own workspace (when logged in) move content items or content item parts around their workspaces through Cut, Copy and Paste actions	☒	☐	
Regular users can, in their own workspace (when logged in) modify, share and classify (through controlled vocabulary, taxonomy and free tags) a content item	☒	☐	
Regular users can, in their own workspace (when logged in) view all content items associated with a tag, add associations between any of their content items, and scroll back and forward all of their content items	☒	☐	
Regular users can, in their own workspace (when logged in) create a content item revision and restore a content item to a previous revision	☐	☒	This functionality is accessible from Advanced Mode, Content item details of i-Publisher.
Regular users can, in their own workspace (when logged in) create / modify / perform a mass operation on all items of / discard / view / filter a selection of content items	☒	☐	
Regular users can, in their own workspace (when logged in) create / modify / remove or add content items to / view a collection of content items	☒	☐	
Regular users can, in their own workspace (when logged in) create / modify / view a controlled vocabulary or taxonomy	☒	☐	
The system allows users to filter content using any combination of context filters: Classification filter, Text mining filter and Search filter	☒	☐	
Users can (in their own workspace) specify which properties for a given content type are indexed and perform full-text search in their own or in the shared content items and annotations	☒	☐	
Users can (in their own workspace) specify the search results to point to the exact location in the content item text where the phrase was found	☒	☐	
Users can (in their own workspace) reset their current	☒	☐	

Simple Confirm Indicators	Success / YES	Failure / NO	Remarks
choice of filters and choose new filter(s) to use			
Users can (in their own workspace) list content items in their workspace and filter them based on meta-data fields	☒	☐	
The system automatically detects the language of each uploaded content item	☒	☐	
The system provides a multilingual user interface	☒	☐	
Users can (in their own workspace) manage automatic classification in terms of defining their own topics, interconnecting them, assigning them to content items, creating new or adjusting existing classification models	☒	☐	
Users can (in their own workspace) specify the text properties of a content item to be used in automatic classification, the classification method and model to be used	☒	☐	
Users can (in their own workspace) specify the text properties of a content item to be used to create summaries, and the summarization method to be used	☐	☒	This functionality is yet to be enabled, since at the moment summarization is still experimental.
Users can (in their own workspace) specify translation model to be used, and the text properties of a content item to be translated and into what languages	☐	☒	This functionality is yet to be enabled, since at the moment machine translation is still experimental.
The system automatically extracts from content items noun phrases, named entities and References (URLs and e-mail addresses)	☒	☐	
Users can specify which text properties of content items are to be processed by the system	☒	☐	
Users can view the text extracted from their documents	☒	☐	

5 TECHNICAL EVALUATION OF PLATFORM COMPONENTS

5.1 *Introduction*

The following sections provide details of the evaluation approach (corpora, method, etc.) followed, the indicators used, and the technical evaluation results for the main ATLAS components:

- Categorization
- Summarization
- Machine translation
- Cross-lingual information retrieval
- CMS and LPCs for the project languages

Where applicable, we also compared the ATLAS components with other baseline relative approaches, and we present the results of our comparative assessment.

5.2 *Categorization*

Evaluation approach

We evaluate the ATLAS categorization module on the Reuters-21578 corpus, which is publicly available and widely accepted as a standard benchmark. In our evaluation the focus is on estimating the correctness of the categorization module, therefore we provide the standard for the categorization task measures - Precision, Recall, and F_1-measure. The ATLAS categorization module is language independent, therefore we perform the evaluation in the English language only.

The experiments were conducted on the ModApte split of Reuters-21578 documents, which is a collection of 12,902 documents for 90 classes, with a fixed splitting between test and training data (3,299 vs. 9,603). The categories were presented as different directories. The set of files (one for each document) associated with the target category, were stored in each directory. The non-labeled documents from the Reuters corpus were stored in the directory "unknown". The document file names were increasing non-repeating numbers for fast document indexing. Two different main directories (test and training) stored the training/testing data.

The corpus was firstly imported in ATLAS in the form of content items from two content types - ReutersTrainingItem (7768) and ReutersTestItem (3019 items). The items in the "unknown" category were excluded from the experiments. Secondly, the imported content items were processed by the English LPC and various categorization models were built. The evaluation procedure compares the manually categorized test items to the automatically suggested categories (folders). The Precision, Recall and F_1-measure are provided for each category in the model. The evaluation is completed with the micro and macro F_1-measures for the whole model.

Indicators

The following formulas are used to average precision, recall, and F_1 across different categories:

	Microaveraging	Macroaveraging
Precision	$\frac{\sum_{i=1}^{\|C\|} TP_i}{\sum_{i=1}^{\|C\|}(TP_i + FP_i)}$	$\frac{\sum_{i=1}^{\|C\|} \frac{TP_i}{TP_i + FP_i}}{\|C\|}$
Recall	$\frac{\sum_{i=1}^{\|C\|} TP_i}{\sum_{i=1}^{\|C\|}(TP_i + FN_i)}$	$\frac{\sum_{i=1}^{\|C\|} \frac{TP_i}{TP_i + FN_i}}{\|C\|}$
F_1	$\frac{2 * \sum_{i=1}^{\|C\|} TP_i}{2 * \sum_{i=1}^{\|C\|} TP_i + \sum_{i=1}^{\|C\|} FP_i + \sum_{i=1}^{\|C\|} FN_i}$	$\frac{\sum_{i=1}^{\|C\|} \frac{2 * TP_i}{2 * TP_i + FP_i + FN_i}}{\|C\|}$

TP_i – true positives, i.e. documents correctly predicted to belong to category c_i
FP_i – false positives, i.e. documents incorrectly predicted to belong to category c_i
FN_i – false negatives, i.e. documents incorrectly predicted not to belong to category c_i

A classifier can be optimized for better precision at the expense of recall, or vice versa, therefore the evaluation results are presented only in terms of F_1 (i.e. a combination of the two).

Evaluation results

We estimate the correctness of four algorithms – Naive Bayesian, Relative Entropy, CFC-modif, and Ensembled. Two types of features (tokens, head nouns) and two feature reduction strategies (tf-idf, chi-square) are used in the evaluation. The results are presented in the table following (figures in bold denote better performance for a given algorithm).

We conclude that:

- Naive Bayesian classifier is more suitable when the number of features is small (less than 1000); The relative entropy and CFC-modif provide better results when the feature space is bigger (more than 4.000 features);

- Reducing the feature space decreases the overall quality of the model but the deviation is acceptable, especially when the feature space is reduced 5 or 10 times;

- All models based on head nouns perform worse than token-based models. The quality of the head nouns models is not significantly worse than the quality of the token-based models; thus head nouns models could be an option for ATLAS installations on modest hardware;

- The quality of models using the chi-2 feature reduction technique is comparable (or slightly better) than the tf-idf top_N feature reduction. However, the complexity of the chi-2 test is $O(n^2)$. One should use the chi2 reduction having in mind that the models are (re)built rarely;

Feature type	Feature reduction	Features Count	Labels	CFC-modif		Relative Entropy		Naive Bayesian		Ensembled	
				Micro F_1	Macro F_1	Micro F_1	Macro F_1	Micro F_1	Macro F_1	Micro F_1	Macro F_1
Token	top_100	39017	1	0,8256	**0,7936**	**0,8763**	0,7154	0,8360	0,6384	0,8524	0,7248
			2	0,8924	**0,8654**	**0,9313**	0,8088	0,8970	0,7126	0,9078	0,8208
			3	0,9232	**0,8915**	**0,9477**	0,8453	0,9222	0,7528	0,9276	0,8638
	top_20	7800	1	0,8068	**0,7170**	**0,8675**	0,6882	0,8343	0,6354	0,8478	0,7183
			2	0,8803	**0,8528**	**0,9245**	0,7734	0,8880	0,6738	0,8994	0,7870
			3	0,9101	**0,8674**	**0,9440**	0,8164	0,9152	0,7320	0,9232	0,8474
	top_10	3900	1	0,7790	0,6817	**0,8632**	0,6536	0,8434	**0,6828**	0,8434	0,6828
			2	0,8632	**0,7687**	**0,9168**	0,7253	0,8890	0,6890	0,8967	0,7677
			3	0,8981	**0,8257**	**0,9380**	0,7806	0,9135	0,7182	0,9182	0,8216
	top_5	1950	1	0,7438	0,5688	**0,8528**	0,6281	0,8374	**0,6345**	0,8374	0,6841
			2	0,8310	0,6811	**0,9064**	**0,7253**	0,8944	0,6811	0,8860	0,7670
			3	0,8719	0,7409	**0,9333**	**0,7627**	0,9155	0,7175	0,9118	0,7907
	top_1	391	1	0,5650	0,3946	0,7795	0,4936	**0,8050**	**0,5312**	0,7365	0,5221
			2	0,6575	0,5052	0,8578	0,5982	**0,8797**	**0,6534**	0,8145	0,6256
			3	0,7123	0,5514	0,8958	0,6583	**0,9052**	**0,6745**	0,8501	0,6541
Head	top_100	4694	1	0,5516	0,3627	0,6408	0,4432	**0,7305**	**0,4985**	0,7166	0,4839
			2	0,6794	0,4871	0,7947	0,5577	**0,8343**	**0,6099**	0,7989	0,6022
			3	0,7343	0,5640	0,8423	0,6311	**0,8729**	**0,6635**	0,8374	0,6545
	top_50	2347	1	0,5461	0,3654	0,6383	0,4371	**0,7329**	**0,4966**	0,7127	0,5028
			2	0,6685	0,4808	0,7934	0,5537	**0,8323**	**0,6228**	0,7962	0,6002
			3	0,7266	0,5487	0,8393	0,6215	**0,8723**	**0,6682**	0,8376	0,6695
	top_20	938	1	0,5115	0,3451	0,6213	0,4044	**0,7305**	**0,4862**	0,7033	0,4822
			2	0,6311	0,4266	0,7155	0,5198	**0,8278**	**0,5566**	0,7866	0,5738
			3	0,6970	0,4985	0,8257	0,5951	**0,8682**	**0,6354**	0,8243	0,6288
	top_5	234	1	0,4003	0,2656	0,5351	0,3175	**0,7087**	**0,4400**	0,6274	0,3472
			2	0,5273	0,3432	0,6331	0,4311	**0,8020**	**0,5315**	0,7264	0,4796
			3	0,5919	0,3989	0,7601	0,4935	**0,8460**	**0,5634**	0,7647	0,5559
Token	chi2_100	6271	1	0,8128	**0,7547**	**0,8712**	0,7244	0,8615	0,7048	**0,8551**	**0,7432**
			2	0,8886	**0,8460**	**0,9316**	0,7933	0,9215	0,7938	**0,9168**	**0,8311**
			3	0,9252	**0,8862**	**0,9530**	0,8271	0,9420	0,8331	**0,9383**	**0,8779**
	chi2_50	3501	1	0,8070	**0,7456**	**0,8601**	0,7075	0,8547	0,6917	0,8581	0,7709
			2	0,8852	**0,8334**	**0,9295**	0,7934	0,9221	0,7837	0,9228	0,8472
			3	0,9242	**0,8897**	**0,9507**	0,8258	0,9460	0,8284	0,9413	0,8691
	chi2_20	1542	1	0,8151	**0,7653**	0,8428	0,6815	**0,8556**	0,7070	0,8614	0,7540
			2	0,8908	**0,8400**	0,9243	0,7720	**0,9307**	0,8021	0,9243	0,8278
			3	0,9236	**0,8787**	0,9473	0,8197	**0,9550**	0,8511	0,9493	0,8727
Head	chi2_100	2479	1	0,5536	0,3754	0,6787	0,4906	0,7290	0,4973	0,7063	0,4712
			2	0,6787	0,4906	0,7848	0,5651	0,8344	0,6085	0,7950	0,5946
			3	0,7356	0,5643	0,8355	0,6406	0,8743	0,6603	0,8351	0,6558
	chi2_50	1882	1	0,5521	0,3720	0,5971	0,4340	**0,7318**	**0,4892**	0,6999	0,4714
			2	0,6757	0,4884	0,7771	0,5439	**0,8354**	**0,6107**	0,7869	0,5876
			3	0,7308	0,5624	0,8259	0,6258	**0,8768**	**0,6628**	0,8266	0,6489
	chi2_20	1102	1	0,5587	0,3854	0,5444	0,3880	**0,7198**	**0,4914**	0,6768	0,4512
			2	0,6736	0,4950	0,7402	0,5148	**0,8339**	**0,6020**	0,7695	0,5756
			3	0,7273	0,5644	0,7938	0,5704	**0,8794**	**0,6612**	0,8132	0,6215

ATLAS categorization module results on the ModApte split on the Reuters-21578 corpus

Comparative assessment

We compared the performance from ATLAS categorization module with some well-known results on the Apte split of Reuters-21578 (see http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-text-classification-1.html for more details):

	Relative Entropy (ATLAS, top 100 tf-idf reduction, one label)	Naive Bayesian (Li and Yang)	Rocchio (Li and Yang)	kNN (Li and Yang)	SVM (Li and Yang)
Micro F_1	87	80	85	86	89
Macro F_1	71	47	59	60	60

Comparison between the best results of ATLAS categorization module and Li and Yang (2003) results on ModApte split of Reuters-21578.

We conclude that all implemented classifiers perform properly and their quality is comparable with the state-of-the-art achievements on the Reuters-21578 corpus.

5.3 *CMS and LPCs for project languages*

Evaluation approach

The ATLAS project had never the intention to develop new LPC tools (e.g. tokenizer, lemmatizer, NP extractor, NER). The idea was to adapt, re-engineer and integrate available (3rd party or in-house developed by the partners) LPC tools and use them either as an external WEB service or within the ATLAS infrastructure. For several of the individual LPC tools for each project language, there are published figures related to the assessment of their performance in terms of Precision (P), Recall (R) and F-measure, which are presented in the subsection "Evaluation of Results" following.

In the scope of the project we evaluated the performance of the LPC tools for all languages (using test corpora of some 70.000 docs in total) in terms of the average processing time for various document size classes (i.e. from 1.000 – 130.000 tokens). The results of this assessment (multi-page report) were presented in D41 deliverable and won't be repeated here.

Furthermore, we performed an additional technical assessment to measure the efficiency of the LPCs and the CMS in terms of the CMS text extraction functionality, and in terms of identified important phrases and named entities. The test corpora we used for this reason consisted of 147 documents, with the following synthesis per project language:

GR	BG	RO	DE	EN	PL	Total
24	28	44	26	21	18	161

Indicators

As mentioned above, most measurements were taken for the "average processing time" indicator. Furthermore, Precision, Recall and F-measure were the indicators used by the majority of the 3rd party LPC tool owners. For our internal technical assessment we used only (P), but in a less strict manner. Using our test documents (147 in total), we assessed the result of certain CMS and LPC functions, by grouping (P) measurements in the following categories:

(P) categories	(P) category value	(P) measurement range	Follow up
Improve	1	0-40%	Corrective measures should be taken
Acceptable	2	41-70%	Improvement should be considered, if feasible
Good	3	71-100%	

In this context, we defined the following Precision indicators:

1. (P) in extracting text from user input
2. (P) in identifying important phrases
3. (P) in identifying Person names
4. (P) in identifying Locations
5. (P) in identifying Organizations

Evaluation results

3rd party published measurements

Bulgarian LPC

	P	R	F
POS tagger	96,58%		
NP recognizer	96,64%	89,03%	92,68%
NER	97,52%	82,67%	89,48%

English LPC

	P	R	F
POS tagger			96,59%
NER	80,41%	78,58%	79,48%

German LPC

	P	R	F
POS tagger	86,05%	83,50%	84,97%

Greek LPC

	P	R	F
POS tagger			84,00%
Named entity recognizer: time expressions	96,62%	92,95%	94,75%
Named entity recognizer: persons	89,06%	85,83%	87,42%
Named entity recognizer: locations	54,19%	51,90%	53,02%
Named entity recognizer: organizations	72,82%	68,99%	70,87%

Polish LPC

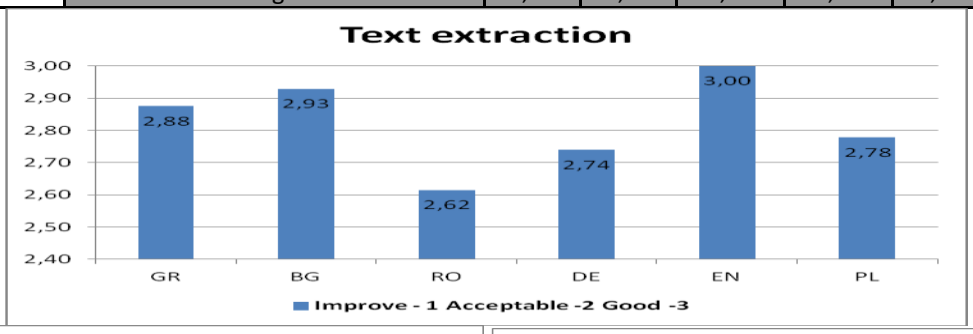
	P	R	F
POS tagger	98,18%	98,16%	98,17%
Shallow parser Spejd (including NP recognition)	97,60%	91,80%	94,61%
NER	83,00%	76,00%	79,00%

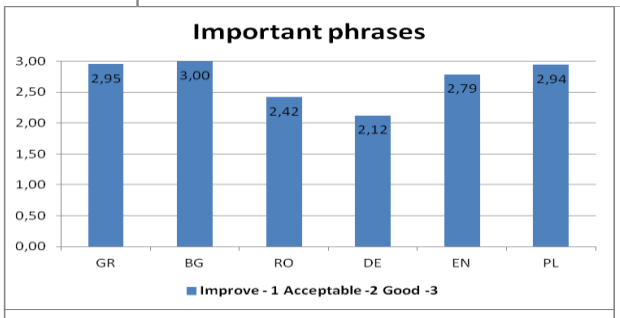
Romanian LPC

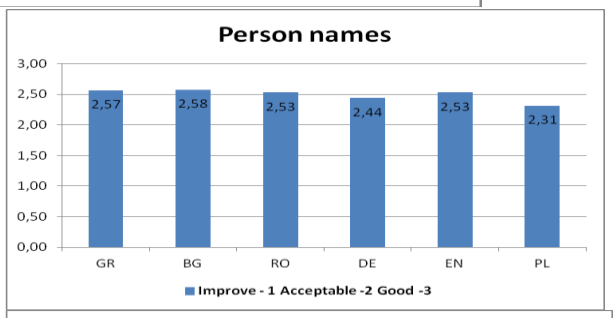
	P	R	F
POS tagger	97,03%		
NP recogniser		98,92%	
NER	79,84%	99,01%	87,45%

Internal assessment based on (P) categories

		GR docs	BG docs	RO docs	DE docs	EN docs	PL docs
(P): Text extraction	# of docs with "Improve – 1"	1	1	1	1	0	1
	# of docs with "Acceptable -2"	1	0	8	4	0	2
	# of docs with "Good -3"	22	27	17	18	18	15
	Normalized Average score	2,88	2,93	2,62	2,74	3,00	2,78
(P): Important phrases	# of docs with "Improve – 1"	0	0	9	3	1	0
	# of docs with "Acceptable -2"	1	0	7	16	2	1
	# of docs with "Good -3"	21	20	27	6	16	17
	Normalized Average score	2,95	3,00	2,42	2,12	2,79	2,94
(P): Person names	# of docs with "Improve – 1"	4	3	3	4	2	0
	# of docs with "Acceptable -2"	2	5	11	6	4	11
	# of docs with "Good -3"	17	18	22	15	11	5
	Normalized Average score	2,57	2,58	2,53	2,44	2,53	2,31
(P): Locations	# of docs with "Improve – 1"	2	1	7	4	2	1
	# of docs with "Acceptable -2"	6	2	21	11	6	9
	# of docs with "Good -3"	15	23	7	7	8	6
	Normalized Average score	2,57	2,85	2,00	2,14	2,38	2,31
(P): Organizations	# of docs with "Improve – 1"	1	0	3	2	0	0
	# of docs with "Acceptable -2"	2	1	6	8	5	4
	# of docs with "Good -3"	21	15	22	15	13	12
	Normalized Average score	2,83	2,94	2,61	2,52	2,72	2,75





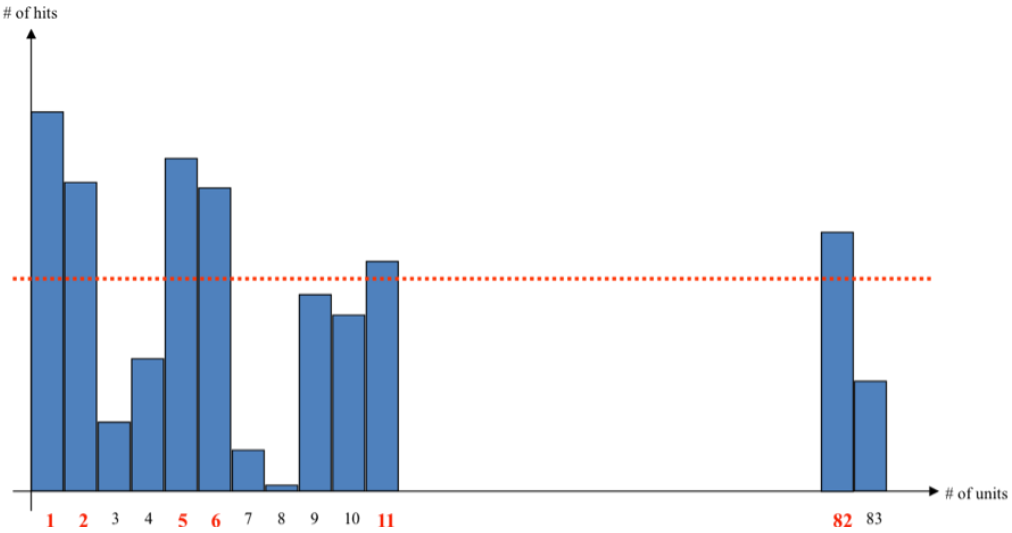


5.4 *Summarization*

Evaluation approach

In the context of the ATLAS summarisation component, the term *short text* refers to a text spanning between ½ - 6 pages. The output of the component is an *excerpt type* summary (i.e. summary that copy clauses from the original text). To assess the performance of the component, we compare this output against a *gold-summary* – a summary produced by humans from the same short text. Therefore, we selected a corpus of short texts in each of the project languages and asked human annotators / summarisers to produce extract type summaries (i.e. sequences of clause IDs). For each short text, in each language, at minimum 3 humans were engaged in order to produce at least 3 summaries (1 summary each). The first of the 3 summaries include 20% of the clauses of the original short text (20% reduction rate), the second summary includes 10% of the original clauses (i.e. selected from the clauses of the 20% reduced summary file), while the third summary includes 5% of the original clauses (i.e. selected from the clauses of the 10% reduced summary file). As a result, we end-up for each short text with minimum 3 (i.e. the minimum number of human summarisers) "20% reduced summaries", minimum 3 "10% reduced summaries" and minimum 3 "5% reduced summaries".

In order to evaluate the performance of the ATLAS summarisation engine / software component, we need to produce measurements for the relevant technical indicators, studying the corpus of annotated (i.e. at clause and marker level) short texts and the respective human-produced gold summaries. To cope with the fact that more than one human produced summaries for the same short text, we need to adopt a method that considers ONE human summary as being gold. The decision was to adopt the Histogram method, by counting the number of times each clause from the original text was mentioned by the human summarisers as belonging to their summaries. In these histograms the sequence of clause numbers is placed on the x-axis and the frequency of mentioning on the y-axis. A sliding horizontal line (threshold) is fixed in this histogram at a position such that the number of units above the line approximates the 20% reduction rate. The respective golden summary is given by all units whose corresponding frequencies were above the threshold (see figure below).



Approximating a gold summary out of a number of human-produced summaries

Two different types of corpora have been used in our experiments: containing clause boundaries and annotation of markers, and containing summaries. The corpora of all languages included short texts of 2 to 4 pages each, from different domains: fairy tales, financial news, political articles, geographical descriptions, etc. The pre-processing chain was launched on each of these texts, producing XML markers, added to the original text, to put in evidence: sentence, clause and token boundaries (these including POS and LEMMA) and markers.

Summaries produced manually were used two fold in our experiments: to calibrate the parameters of the discourse parser and to finally evaluate the whole summarisation chain and the automatically produced summary, as the final output. As mentioned already, the summaries included a list of clause IDs, indicating the clauses considered by the human subjects to be part of the summary.

Indicators

Given a short text file, we annotate it at clause and marker level (input file), we create a human generated gold summary file through 20% reduction at clause level (gold_sum_file), and we produce an automatically generated summary file through the ATLAS summarisation engine (auto_sum_file). In this context, the following indicators were used for the technical evaluation of the summarisation component:

- SUM_PREC: the fraction of clauses in the auto_sum_file that are included also in the gold_sum_file – summarisation precision.

$$SUM\_PREC = \frac{\text{Number of clauses in auto\_sum\_file,also included in gold\_sum\_file}}{\text{total number of clauses in auto\_sum\_file}}$$

- SUM_REC: the fraction of clauses in the gold_sum_file that are also included in the auto_sum_file – summarisation recall

$$SUM\_REC = \frac{\text{Number of clauses in auto\_sum\_file, also included in gold\_sum\_file}}{\text{total number of clauses in gold\_sum\_file}}$$

- SUM_FM: the weighted harmonic mean of summarisation precision and recall – summarisation F-measure.

$$SUM\_FM = \frac{2 * \text{SUM\_PREC} * \text{SUM\_REC}}{\text{SUM\_PREC} + \text{SUM\_REC}}$$

Evaluation results

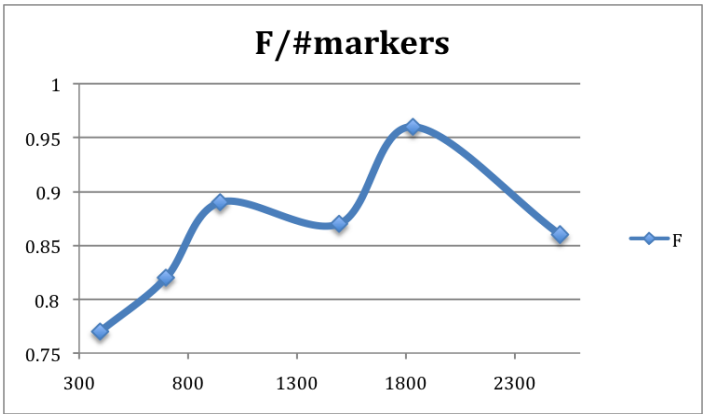
In the following tables we present the dimension of the evaluation corpora and the clause segmentation evaluation results (by comparing the number of boundaries) for each of the languages under experiment. In the last column, the evaluation data represent averages over all languages.

Language	BG	DE	EN	GR	PL	RO	TOTAL
# sentences	2.749	1.375	2.246	1.055	1.096	1.571	10.092

# tokens	51.116	31.839	53.504	30.207	21.377	47.016	235.059
# clauses	6.468	2.726	4.880	2.778	2.574	3.720	23.146
# markers	2.507	396	1.832	1.493	698	947	7.873

Language		BG	DE	EN	GR	PL	RO	AVG
Evaluation	SUM_PREC	0,97	0,93	0,98	0,90	0,89	0,91	0,93
	SUM_REC	0,77	0,66	0,94	0,84	0,97	0,88	0,84
	SUM_FM	0,86	0,77	0,96	0,87	0,82	0,89	0,86

When comparing the quantitative data with the evaluation results, there seems to be evidence of a number of correlations. For instance, it is clear that the dimension of the corpus (#tokens, #clauses, #markers, etc.) influence the quality of the segmenter. If we plot on the same graph the F-measures of all languages, in correlation with the number of markers of their corresponding corpora, the following figure is produced. This figure reveals that there is a certain monotonicity tendency. However, it can also be noticed that languages like GR and BG (whose F-measures are lower than the interpolation over all languages, marked with a thin line) seem to need more data for equivalent segmentation quality.



The correlation between the #markers and F-measure

Comparative assessment

Using the same technical indicators (precision, recall, FM), we attempted a comparative assessment between the ATAL summariser and other well-known summarisers. We considered the following two summarisers:

1. Open Text Summarizer (OTS): OTS considers that the important ideas in an article repeat in the same words the main subject of the article, expressed in rather technical terms. It is seldom used as a benchmark for other summarization systems.
2. LexRank: LexRank (Erkan and Radev, 2004) computes the relative importance of textual units and sentences based on the concept of eigenvector centrality in a graph representation of sentences. Instead of passing words to the summarizer, we were passing sequences of numbers – token IDs, NP IDs, NE IDs. In this way we made the input to the LexRank summarizer language independent

The figures in the following Table are computed by comparing occurrences of IDs of clauses in the test against those in the gold summaries.

Language		BG	DE	EN	GR	PL	RO	AVG
ATLAS summarizer	P (H)	**0,19**	**0,23**	**0,27**	0,23	0,17	0,22	**0,22**
	R (H)	**0,29**	**0,44**	**0,41**	**0,41**	**0,36**	**0,32**	**0,37**
	F (H)	**0,23**	**0,30**	**0,32**	**0,29**	**0,23**	**0,25**	**0,27**
OTS summarizer	P (H)	0,16	0,19	0,24	**0,27**	0,19	**0,29**	**0,22**
	R (H)	0,25	0,20	0,22	0,33	0,21	0,06	0,21
	F (H)	0,19	0,20	0,23	0,27	0,20	0,10	0,20
LexRank summarizer	P (H)	0,15	**0,23**	**0,27**	0,24	**0,24**	0,21	0,21
	R (H)	0,18	0,25	0,25	0,22	0,24	0,22	0,18
	F (H)	0,16	0,24	0,26	0,23	0,22	0,21	0,19

The H's appearing in parenthesis after the three evaluation measures (precision, recall and F-measure) signify that the gold data used for comparison have been approximated out of the ones indicated by humans, by using the histogram method described above. As can be noticed (the best values are marked in bold), our summarizer performs better globally (in terms of F-scores) than the other two methods.

5.5 *Machine translation*

Evaluation approach

The evaluation of machine translation (MT) systems is still an open research issue. There are two main directions in their evaluation:

1. The human evaluation following criteria like intelligibility, readability, time used for correction;
2. The automatic evaluation based on measures like BLEU, TER, METEOR.

Given the number and the diversity of language pairs covered by the ATLAS system, a manual evaluation was impossible. Therefore, we decided to use for the evaluation the widely used automatic metric BLEU. This gave us the possibility to make an approximate comparison with gold- systems. We will explain in the next subsections why we consider this only an "approximate" comparison.

The ATLAS-MT-engine contains domain specific models for 13 domains. The adaptation was done by injecting a small in-domain parallel corpus into the larger and more general JRC-Acquis corpus. For evaluation purposes, we isolated before training **5% of sentences from the in-domain corpus**[1], as test data. The test-data was not involved in the training process.

Indicators

As explained above, we decided to use the BLEU metric as our main indicator. In addition, we decided to use also as indicator the number of non-translated words in the test-set mentioned

[1] in-domain test sentences are sentences belonging to the same domain as the training data. For example, if training is based on a "Law" domain and the testing is done with sentences from the same domain, than these sentences are referred to as "in-domain test sentences". On the other hand, if the testing is done with sentences from a "Computer Science" domain, then is test set is referred to as "out of domain test sentences"

above (as the ATLAS MT component follows a corpus-based approach, we were interested to see how broad is the language coverage).

1 - MT_BLEU

BLEU (bilingual evaluation understudy), one of the evaluation scores applied most frequently for MT evaluation, measures the number of n-grams of different lengths of the system output that appear in a set of references. Although criticized more recently, it is still important to calculate the BLEU score for comparison reasons, as for many previous developed systems it is the only evaluation measure available. The BLEU score is computed according to the following formula:

$$BLEU = BP * exp(\sum_{n=1}^{N} \frac{1}{N} log(p_n)) \quad (1)$$

where N is the maximum n-gram size and the brevity penalty, BP, is calculated as:

$$BP = min(1, e^{1-\frac{r}{c}}) \quad (2)$$

In Formula (2), c is the length of the corpus of hypothesis translations and r is the effective reference corpus length. The value for r is calculated as the sum of the single reference translation from each of the set which is closest to the hypothesis translation.

The n-gram precision p_n is calculated as the sum over the matches for every hypothesis sentence S in the complete corpus C, as:

$$p_n = \frac{\sum_{S \in C} \sum_{ngram \in S} count_{matched}(ngram)}{\sum_{S \in C} \sum_{ngram \in S} count(ngram)} \quad (3)$$

For the evaluation with BLEU, we used the twelfth version of the NIST/BLEU implementation provided by www.itl.nist.gov.

2 - MT_WDi:

This is count the percentage of non translated words for each project language pair (i=1..lang_pairs). We omit here the Name entities.

Evaluation results

The evaluation of machine translation engine cannot be done overall but per domain, as we have separate translation models per domain. The size of the in-domain training data varies from one domain to another. Overall the ATLAS MT system relies mainly on 390 translation subsystems (2 models per language pair x 15 language pairs x 13 domains). In the following tables we present the results for MT_BLEU for two domains, one of scientific nature (Computer Science) and one related to humanities (Politics).

1 - MT_BLEU

POLITICS

Language Pair	BLEU score	#sentences in-domain corpus	#sentences out-domain corpus	#sentences test set
BG-DE	8.70	1914	344997	100
BG-EL	28.14	52930	515072	2750
BG-RO	16.53	59371	241670	3000
BG-EN	39.36	56796	306767	3000
BG-PL	7.05	1981	367523	100
RO-DE	8.88	2261	324448	100
RO-EL	4.41	53613	159417	2750
RO-BG	19.09	59371	241670	3000
RO-EN	39.12	64329	336455	3351
*RO-PL	6.12	1912	362321	100
EL-RO	13.91	53613	159417	2750
EL-EN	36.68	51564	592923	2750
EL-BG	31.76	52930	515072	2750
EL-PL	5.54	1963	641689	100
EL-DE	12.01	1912	719960	100
*PL-RO	9.76	1914	362321	100
PL-EN	14.15	1889	1183516	100
PL-DE	4.00	1930	1179492	100
PL-EL	4.45	1963	641689	100
PL-BG	14.62	1981	367523	100
EN-DE	19.75	1988	1199447	100
EN-EL	30.81	51564	592923	2750
EN-RO	39.75	64329	336455	3351
EN-BG	0.31	56796	306767	3000
EN-PL	8.97	1889	1183516	100
DE-EN	24.35	1988	1199447	100
DE-PL	13.72	1930	1179492	100
DE-RO	4.64	2261	324448	100
DE-EL	10.82	1912	719960	100
DE-BG	13.32	1914	344997	100

BLEU score for domain “Politics”

COMPUTING

Language Pair	BLEU score	#sentences in-domain corpus	#sentences out-domain corpus	#sentences test set
BG-DE	16.33	2536	344997	125
BG-EL	20.90	2352	515072	100
BG-RO	17.26	3061	241670	150
BG-EN	27.29	2642	306767	125
BG-PL	15.60	2085	367523	100
RO-DE	7.79	4836	324405	250
RO-EL	20.50	2046	159417	100
RO-BG	16.72	3061	241670	150
*RO-EN	5.21	5133	336455	250
RO-PL	3.41	3566	362321	175
EL-RO	21.23	2046	159417	100
*EL-EN	7.20	3963	592923	200
EL-BG	19.19	2352	515072	100
*EL-PL	3.43	1963	641689	125
EL-DE	11.53	4439	719960	255
PL-RO	1.36	3566	362321	175
*PL-EN	3.6	9013	1183516	500
*PL-DE	2.99	8384	1179492	425
*PL-EL	2.75	1963	641689	125
PL-BG	14.59	2085	367523	100
*EN-DE	30.21	40459	1199447	2125
*EN-EL	7.75	3963	592923	200
*EN-RO	4.45	5133	336455	250
EN-BG	19.12	2642	306767	125
*EN-PL	2.06	9013	1183516	500
*DE-EN	2.70	40459	1199447	2125
*DE-PL	0.85	8384	1179492	425
DE-RO	6.07	4836	324405	250
DE-EL	11.54	4439	719960	255
DE-BG	14.49	2536	344997	125

BLEU score for domain "Computing"

2 - MT_WDi

This evaluation was done manually as there is no possibility to automatically identify the language of isolated words. Given the diversity of language pairs and the language competence of available personal resources, we decided to measure this indicator for English translation. This decision is motivated also by the fact that usually the parallel corpora including English are the richest (i.e. they have the broadest coverage). In this way our measurements give an

upper margin for this indicator. We choose a domain in which the number of English neologism is less frequent: Politics

Language pair	Nr. of words in the test data	MT_WDi
BG-EN	85273	0,08%
DE-EN	3980	0,57%
EL-EN	81710	0,14
PL-EN	2169	2,4 %
RO-EN	86094	0,05%

WDi for translations into English, domain Politics

Comparative assessment

Given the fact that corpus-based MT-systems are extremely sensible to the type and amount of training data, only an approximate comparison is possible. The very last results published by META-NET in 2012 rely on tests done on in-domain training data from JRC Acquis communautaire. These results are presented in the table following.

	Zielsprache – Target language																					
	EN	BG	DE	CS	DA	EL	ES	ET	FI	FR	HU	IT	LT	LV	MT	NL	PL	PT	RO	SK	SL	SV
EN	–	40.5	46.8	52.6	50.0	41.0	55.2	34.8	38.6	50.1	37.2	50.4	39.6	43.4	39.8	52.3	49.2	55.0	49.0	44.7	50.7	52.0
BG	61.3	–	38.7	39.4	39.6	34.5	46.9	25.5	26.7	42.4	22.0	43.5	29.3	29.1	25.9	44.9	35.1	45.9	36.8	34.1	34.1	39.9
DE	53.6	26.3	–	35.4	43.1	32.8	47.1	26.7	29.5	39.4	27.6	42.7	27.6	30.3	19.8	50.2	30.2	44.1	30.7	29.4	31.4	41.2
CS	58.4	32.0	42.6	–	43.6	34.6	48.9	30.7	30.5	41.6	27.4	44.3	34.5	35.8	26.3	46.5	39.2	45.7	36.5	43.6	41.3	42.9
DA	57.6	28.7	44.1	35.7	–	34.3	47.5	27.8	31.6	41.3	24.2	43.8	29.7	32.9	21.1	48.5	34.3	45.4	33.9	33.0	36.2	47.2
EL	59.5	32.4	43.1	37.7	44.5	–	54.0	26.5	29.0	48.3	23.7	49.6	29.0	32.6	23.8	48.9	34.2	52.5	37.2	33.1	36.3	43.3
ES	60.0	31.1	42.7	37.5	44.4	39.4	–	25.4	28.5	51.3	24.0	51.7	26.8	30.5	24.6	48.8	33.9	57.3	38.1	31.7	33.9	43.7
ET	52.0	24.6	37.3	35.2	37.8	28.2	40.4	–	37.7	33.4	30.9	37.0	35.0	36.9	20.5	41.3	32.0	37.8	28.0	30.6	32.9	37.3
FI	49.3	23.2	36.0	32.0	37.9	27.2	39.7	34.9	–	29.5	27.2	36.6	30.5	32.5	19.4	40.6	28.8	37.5	26.5	27.3	28.2	37.6
FR	64.0	34.5	45.1	39.5	47.4	42.8	60.9	26.7	30.0	–	25.5	56.1	28.3	31.9	25.3	51.6	35.7	61.0	43.8	33.1	35.6	45.8
HU	48.0	24.7	34.3	30.0	33.0	25.5	34.1	29.6	29.4	30.7	–	33.5	29.6	31.9	18.1	36.1	29.8	34.2	25.7	25.6	28.2	30.5
IT	61.0	32.1	44.3	38.9	45.8	40.6	26.9	25.0	29.7	52.7	24.2	–	29.4	32.6	24.6	50.5	35.2	56.5	39.3	32.5	34.7	44.3
LT	51.8	27.6	33.9	37.0	36.8	26.5	21.1	34.2	32.0	34.4	28.5	36.8	–	40.1	22.2	38.1	31.6	31.6	29.3	31.8	35.3	35.3
LV	54.0	29.1	35.0	37.8	38.5	29.7	8.0	34.2	32.4	35.6	29.3	38.9	38.4	–	23.3	41.5	34.4	39.6	31.0	33.3	37.1	38.0
MT	72.1	32.2	37.2	37.9	38.9	33.7	48.7	26.9	25.8	42.4	22.4	43.7	30.2	33.2	–	44.0	37.1	45.9	38.9	35.8	40.0	41.6
NL	56.9	29.3	46.9	37.0	45.4	35.3	49.7	27.5	29.8	43.4	25.3	44.5	28.6	31.7	22.0	–	32.0	47.7	33.0	30.1	34.6	43.6
PL	60.8	31.5	40.2	44.2	42.1	34.2	46.2	29.2	29.0	40.0	24.5	43.2	33.2	35.6	27.9	44.8	–	44.1	38.2	38.2	39.8	42.1
PT	60.7	31.4	42.9	38.4	42.8	40.2	60.7	26.4	29.2	53.2	23.8	52.8	28.0	31.5	24.8	49.3	34.5	–	39.4	32.1	34.4	43.9
RO	60.8	33.1	38.5	37.8	40.3	35.6	50.4	24.6	26.2	46.5	25.0	44.8	28.4	29.9	28.7	43.0	35.8	48.5	–	31.5	35.1	39.4
SK	60.8	32.6	39.4	48.1	41.0	33.3	46.2	29.8	28.4	39.4	27.4	41.8	33.8	36.7	28.5	44.4	39.0	43.3	35.3	–	42.6	41.8
SL	61.0	33.1	37.9	43.5	42.6	34.0	47.0	31.1	28.8	38.2	25.7	42.3	34.6	37.3	30.0	45.9	38.2	44.1	35.8	38.9	–	42.7
SV	58.5	26.9	41.0	35.6	46.6	33.3	46.6	27.4	30.9	38.9	22.7	42.0	28.2	31.0	23.7	45.6	32.2	44.2	32.7	31.3	33.5	–

BLEUGold-standard measurements as in "The German Language in the digital age"- A. Burchardt, M. egg, K. eichler, B. Krenn, J. Kreutel, A. Leßmöllmann, g. Rehn, M. stede, H. Uszkoreit, M. Volk., METa-NET White papers , Springer 2012

Language Pair	Computing	Politics	Gold
BG-DE	16.33	8.70	38.7
BG-EL	20.90	28.14	34.5
BG-RO	17.26	16.53	36.8
BG-EN	27.29	39.36	61.3
BG-PL	15.60	7.05	35.1
RO-DE	7.79	8.88	38.5
RO-EL	20.50	4.41	35.6
RO-BG	16.72	19.09	33.1
*RO-EN	5.21	39.12	60.8
RO-PL	3.41	6.12	35.8
EL-RO	21.23	13.91	37.2
*EL-EN	7.20	36.68	59.5
EL-BG	19.19	31.76	32.4
*EL-PL	3.43	5.54	34.2
EL-DE	11.53	12.01	43.1
PL-RO	1.36	9.76	38.2
*PL-EN	3.6	14.15	60.8
*PL-DE	2.99	4.00	40.2
*PL-EL	2.75	4.45	34.2
PL-BG	14.59	14.62	31.5
*EN-DE	30.21	19.75	46.8
*EN-EL	7.75	30.81	41.0
*EN-RO	4.45	39.75	49.0
EN-BG	19.12	0.31	40.5
*EN-PL	2.06	8.97	49.2
*DE-EN	2.70	24.35	53.6
*DE-PL	0.85	13.72	30.2
DE-RO	6.07	4.64	30.7
DE-EL	11.54	10.82	32.8
DE-BG	14.49	13.32	26.3

Gold-standard BLEU and values for 2 domains in ATLAS system

We observe that the scores obtained for the ATLAS system are below the gold scores. This is not an indication for the quality of the systems as in fact one evaluation exercise cannot be replicated. Our tests are performed on different test sets, on different domains as the Gold Standard.

The systems marked with "*" are still in development, in the sense that the sentence-alignment of the "in-domain data" has to be revised. The wrong automatic sentence alignment has a major influence on the translation quality.

5.6 *Cross-lingual information retrieval*

Evaluation approach

The approach we followed is based on evaluating the technical performance of the CLIR component in isolation, through importing document files in all project languages in a directory structure. When imported, each file generates a routing message, which in turn is converted into an RDF Index storage event or query. It makes no sense to use statistical metrics (e.g. precision, recall, etc.) for the technical assessment, as the results would be deterministic and the same each time we would execute the same query on the same test corpus. Therefore, we decided to measure the time element involved in the different CLIR processes, as this is what the user experiences when using the CLIR through the ATLAS applications. Hence, throughout the tests the processing time was logged in milliseconds, for different internal processes:

- Process and index a single text document, when imported into the system.
- Process and index a batch archive file (containing multiple single documents), when imported into the system.
- Process and input query with 2 search-terms and return the results.
- Process and input query with 4 search-terms and return the results.

For our measurements we used a test corpus of 528 text document files from all project languages. The average size of the documents was 72,4 KB; the largest was 4,1 MB big and the smallest 1,2 KB. For testing the batch processing we used a batch file containing 40 files; the largest was 527,9 KB, while the smallest was 149,2 KB.

For the 2 search-terms we defined 120 queries in all project languages, while for the 4 search-terms we defined 110 queries.

Example of 2 search-terms queries in the project languages

```
<search query="@content:Modern @content:blogs" rows="20" />
<search query="@content:способността @content:развода" rows="20" />
<search query="@content:Glockenspiel @content:Warren" rows="20" />
<search query="@content:δυσκολία @content:Κατηγοριών" rows="20" />
<search query="@content:polepszenie @content:Ćwiczebne" rows="20" />
<search query="@content:subiecti @content:artizanilor" rows="20" />
```

Example of 4 search-terms queries in the project languages

```
<search query="@content:новинарските @content:Хърватският @content:Оператор @content:Загуба" rows="20"/>
<search query="@content:Zeitungen @content:Reduzieren @content:Halb @content:gute" rows="20"/>
<search query="@content:Έξυπνοι @content:ευρέως @content:Λισσαβώνα @content:αρχαρίων" rows="20"/>
<search query="@content:ironical @content:Propaganda @content:seeker @content:Opportunity" rows="20"/>
<search query="@content:logiki @content:Glacjalnie @content:Pasek @content:ciężarówka" rows="20"/>
<search query="@content:injurie @content:Controlul @content:Guyana @content:gust" rows="20"/>
```

The number of queries per language for each search type was the following:

Search type	BG	DE	GR	EN	PL	RO	Total
2-terms	21	20	21	19	18	21	**120**
4-terms	18	17	21	20	17	17	**110**

Since we measured time, we provide below some factual figures of the environment in which we run the experiments:

- Memory: total 3.983.164 KB , free 470.132 KB, Cached 462.552 KB
- CPU: Intel Core2 Quad CPU Q9650 @ 3.00GHz , with 4 CPU cores
- O/S: Linux 3.5.0-22-generic #34-Ubuntu SMP x86_64 GNU/Linux

Indicators

As explained above, we decided to measure the time element involved in the different CLIR processes; for this, we used the following indicators:

- CLR_INS: time needed to process and index a single text document, when imported into the system.
- CLR_BINS: time needed to process and index a batch archive file (containing multiple single documents), when imported into the system.
- CLR_EXEC2: time needed to render the results of a search, using a query with 2 input terms.
- CLR_EXEC4: time needed to render the results of a search, using a query with 4 input terms.

Evaluation results

We present below the figures we measured for the different CLIR indicators in terms of both the required time for processing a single document or query and the time for all documents or queries. For the single document / query our target was < 1 sec and this was achieved in all cases.

	CLR_INS	CLR_BINS	CLR_EXEC2	CLR_EXEC4
Number of docs	528	40	524	524
Number of queries	1	1	120	110
Average time for single doc / query (in secs)	0,063	0,030	0,367	0,619
Total time (in secs)	33,46	0,308	44,055	68,176