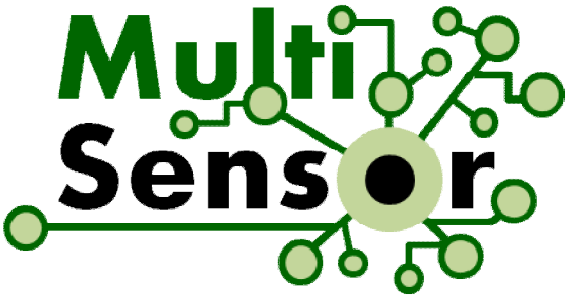

MULTISENSOR

Mining and Understanding of multilinguaLcontenT for Intelligent Sentiment Enriched coNtext and Social Oriented inteRpretation

FP7-610411

D2.3

Advanced techniques for text analysis, machine translation and concept extraction

Dissemination level:	Public
Contractual date of delivery:	Month 23, 30/09/2015
Actual date of delivery:	Month 24, 09/10/2015
Workpackage:	WP2 Multilingual and Multimedia Content Extraction
Task:	T2.2 Named entity extraction workflows T2.3 Concept extraction from text T2.4 Concept linking and relations T2.5 Audio transcription and analysis T2.6 Multimedia concept and event detection T2.7 Machine translation
Type:	Report
Approval Status:	Final
Version:	1.1
Number of pages:	60
Filename:	D2.3_AdvancedTechniques_2015-10-09_v1.1.pdf

Abstract

This deliverable is the second report on techniques for text analysis, machine translation, speech recognition, and concept extraction. It expands the work done for the deliverable D2.2 (basic techniques in all modules), continues the description of basic techniques, presents the progress in the development of all modules, discusses the advanced techniques applied so far, and reports the evaluation results.

Co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	02/09/2015	Draft TOC	R.Busch (LT)
0.2	18/09/2015	Contributions from Linguatec to sections 5 and 7	B. Vaisman, R.Busch (LT)
0.3	21/09/2015	Contributions from CERTH to section 6	D. Liparas, A. Moumtzidou, S. Vrochidis (CERTH)
0.4	22/09/2015	Contributions from Linguatec to section 2	G.Thurmair (LT)
0.5	24/09/2015	First consolidated draft	R.Busch (LT)
0.6	25/09/2015	Contribution from UPF to sections 3 and 4	G. Casamayor (UPF)
0.7	26/09/2015	Second consolidated draft	R. Busch (LT)
0.8	29/09/2015	Internal Review	A.Moumtzidou (CERTH)
0.9	02/10/2015	Contribution from CERTH and Linguatec	D. Liparas (CERTH), G.Thurmair (LT)
0.95	03/10/2015	Third consolidated draft	R. Busch (LT)
1.0	08/10/2015	Final version	G. Casamayor, S. Mille (UPF)

Author list

Organization	Name	Contact Information
Linguatec	Reinhard Busch	r.busch@linguatec.de
Linguatec	Gregor Thurmair	g.thurmair@linguatec.de
Linguatec	Boris Vaisman	b.vaisman@linguatec.de
CERTH	DimitrisLiparas	dliparas@iti.gr
CERTH	Anastasia Moumtzidou	moumtzid@iti.gr
CERTH	StefanosVrochidis	stefanos@iti.gr
UPF	Gerard Casamayor	gerard.casamayor@upf.edu
UPF	Simon Mille	simon.mille@upf.edu

Executive Summary

This deliverable reports on advanced techniques for text analysis, machine translation, speech recognition, and concept extraction in the MULTISENSOR project. It presents the overall progress in WP2, and it in particular concentrates on achievements done after the submission of D2.2, in which the basic techniques and approaches for all WP2 modules have been described.

The document describes in detail all modules in WP2, their approaches, components, andresources. Furthermore, the processof the creation of training and testing datasets is described, the evaluation approaches and tools areexplained, and the evaluation results presented. Specifically the following components are presented:

a) named entities recognition(T2.2)
b) concept extraction from text and concept linking &relation extraction(T2.3/T2.4)
c) speech recognition (T2.5)
d) concept extraction from images and video (T2.6)
e) machine translation (T2.7)
f) language identification from text (not defined in the DoW)

The deliverable D2.3 presents the work done during the second year of the project, and it contributes to the achievement of milestones MS3 and MS4 (first and second prototype).

Abbreviations and Acronyms

AP	Average Precision
API	Application programming interface
ARPA	Advanced Research Projects Agency
AS	Automatic Summarization
ASR	Automatic speech recognition
BLEU	Bilingual Evaluation Understudy
BoW	Bag of Words
CMLLR	Constrained Maximum Likelihood Linear Regression
CSV	Comma-separated values
DGT	Directorate-General for Translation (EU)
FV	Fisher Vector
G2P	Grapheme to Phoneme
GATE	General Architecture for Text Engineering
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IE	Information extraction
IR	Information retrieval
JPEG	Joint Photographic Experts Group (Image file format)
JRC	Joint Research Centre (EU)
JSON	JavaScript Object Notation
JSON-LD	JSON for Linked Data
JSONP	JSON (JavaScript Object Notation) with padding
LDA	Linear Discriminant Analysis
LOD	Linked Open Data
LVCSR	Large Vocabulary Continuous Speech Recognition
MFCC	Mel Frequency Cepstral Coefficients
MPEG	Moving Picture Experts Group (Audio/visual file format)
NE(R)	Named Entities (Recognition)
NED	Named Entities Disambiguation
NIF	NLP Interchange Format
NIST	National Institute of Standards and Technology
NLG	Natural Language Generation
NLP	Natural Language Processing
OCR	Optical character recognition
OOV	Out of Vocabulary
OWL	Web Ontology Language
PCA	Principal Component Analysis
PDF	Portable Document Format
PLP	Perceptual Linear Prediction
RBMT	Rule-based Machine Translation
RDF	Resource Description Format
REST	Representational State Transfer

RSS	Really Simple Syndication (or Rich Site Summary)
SIFT	Scale-invariant feature transform
SIN	Semantic Indexing Task
SME	Small and medium-sized enterprises
SMT	Statistical Machine Translation
SRL	Semantic Role Labelling
SURF	Speeded Up Robust Features
SVM	Support vector machine
TRECVID	TREC Video Retrieval Evaluation
UC1	Use Case 1
UC2	Use Case 2
URI	Uniform Resource Identifier
UIMA	Unstructured Information Management Applications
UVI	Unified Verb Index
VLAD	Vector of Locally Aggregated Descriptors
VTLN	Vocal Tract Length Normalization
WER	Word Error Rate
WP	Work Package
WSD	Word Sense Disambiguation
XML	Extensible Markup Language

Table of Contents

1 INTRODUCTION

This deliverable reports on the work done in WP2 of the MULTISENSOR project during the second project year. The objective of WP2 is to extract knowledge from multimedia input data, and present it in a way that later components can operate on them.

The current report comprises all tasks of WP2, except of T2.1 that was successfully completed in month 6 of the project and described in D2.1 (Empirical study on media monitoring and internationalisation resources). Accordingly, it describes the work done in tasks T2.2 (Named entity extraction workflows), T2.3 (Concept extraction from text), T2.4 (Concept linking and relations), T2.5 (Audio transcription and analysis), T2.6 (Multimedia concept and event detection), and T2.7 (Machine translation). Additionally, we describe the language identification component, which was not foreseen in the DoW but turned out to be important for efficient text processing, and as such had not been included so far in any WP2 tasks.

All mentioned WP2 tasks contribute to the milestones MS3and MS4 of the project (first and second prototype of the MULTISENSOR system). They correspond to the second year (Y2) activities A2.1 to A2.6, described in the project roadmap D7.1 as shown in Figure 1.

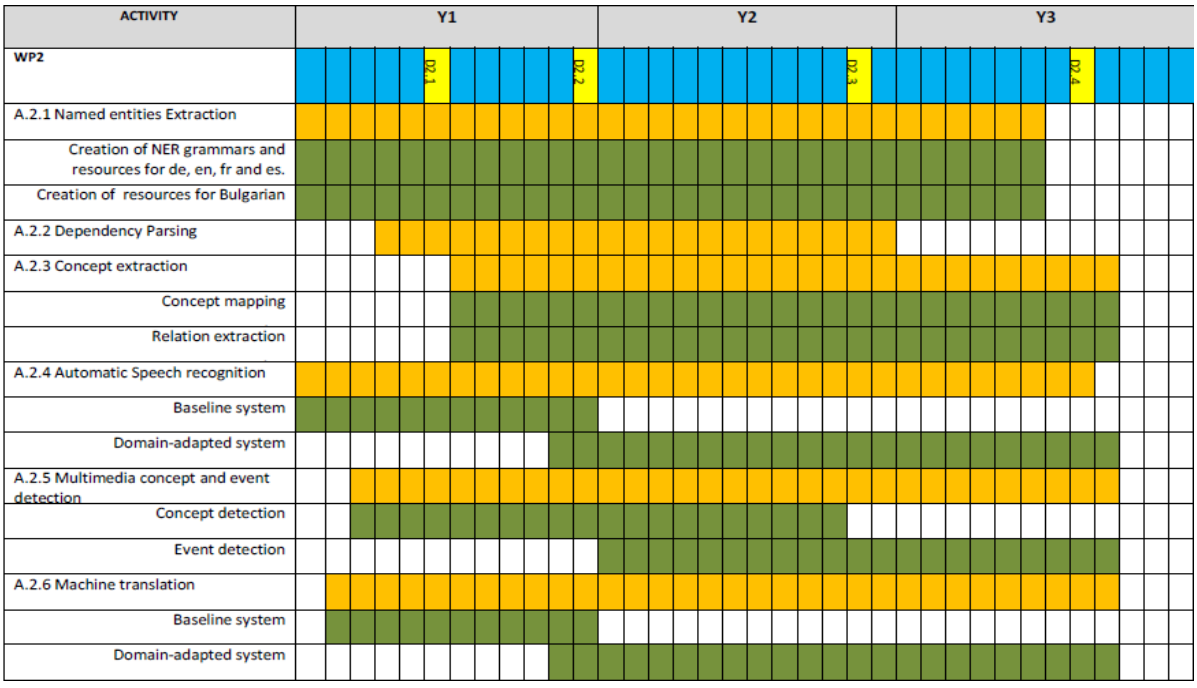

Figure 1: WP2 activities and timeline

In this deliverable, we report on each task in a different section (Sections 2, 3, 4, 5, 6, and 7). In section 8 we additionally describe the language identification module that was released during the second year of the project. The introductory section gives an overview of the information extraction pipelines and the general architecture of WP2, while in section 9 we draw some conclusions about the work done in WP2, with special emphasis on work done after the deliverable D2.2, which was submitted in the month 12.

1.1 Architecture of the content extraction module

The objective of WP2 ‚Multilingual and Multimedia Content Extraction' is to extract knowledge from multimedia input data (audio, text, video, image). In WP2, this is done by consecutively running a series of content analysis and extraction services, as illustrated in Figure 2.

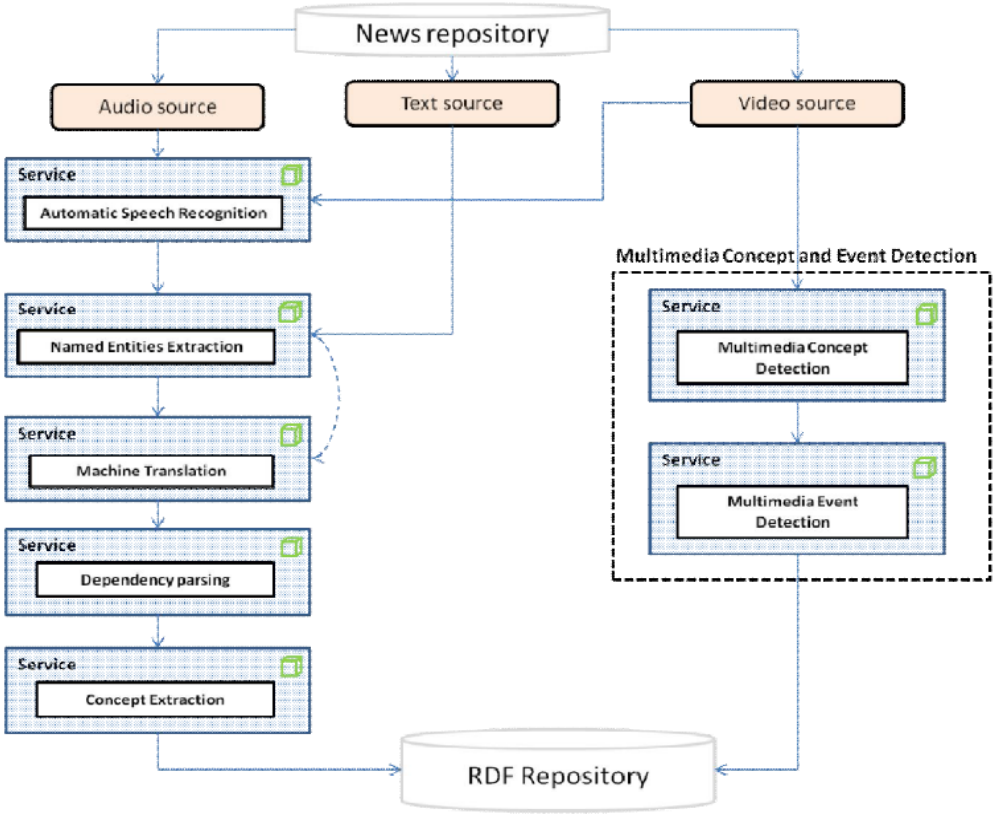

Figure 2: WP2 pipeline

There are two main analysis pipelines in WP2: text analysis and video/image analysis. Besides the original text sources, the subject of the text analysis pipeline are speech-to-text transcriptions from audio and video input data. The recognised and extracted knowledge from both extraction pipelines is presented in a structured way (as RDF statements) and stored in the RDF repository. All later components (content integration and retrieval, semantic reasoning, and abstractive summarisation) can then operate on the information extracted and organised in this way.

1.2 Formats and services

All WP2 modules are deployed as REST[1] web services. They communicate with each other via public APIs. More information can be found in D7.1 ‚Roadmap towards the implementation of MULTISENSOR platform', D7.2 ‚Technical requirements and architecture design', as well as in D2.3 ‚Basic techniques for speech recognition, text analysis, and concept detection'.

[1] cf. http://en.wikipedia.org/wiki/Representational_state_transfer

2 NAMED ENTITIES RECOGNITION

The named entities recognition in MULTISENSOR component is a linguistic analyser using knowledge-based technology. The architecture has been described in deliverable ‚D2.2 MULTINSENSOR Basic techniques', and repeated here in Figure 3.

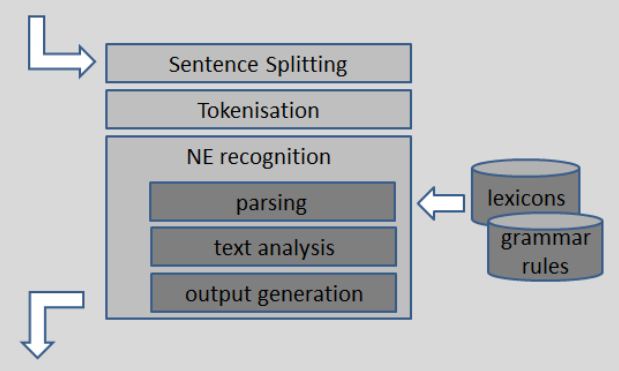

Figure 3: NE recognition pipeline

This architecture has been modified in the reporting period:

- **Integration** into the CEP required a change in the component interface. The input now is given in the CEP workbook format[2], and also the output is supposed to be provided in this format, not in one of the formats supported before (mark-up, or JSON).
- The **components** themselves had to be adapted to compute, and preserve, the character offsets of the respective tokens.

In addition to these changes, the NER component itself underwent major extensions and improvements, mainly in its language **resources**:

- the lexicons were adapted to the three domains / application scenarios tackled in MULTISENSOR, resulting in several thousand additional lexicon entries
- the grammars were extended beyond a pure lexicon-based level to a more context-sensitive approach by which content indicators are used to determine an NE and its type

As for **testing**, test corpora were created, taken from documents of the current repositories, and manually annotated as a kind of ‚gold standard' in a bootstrapping fashion. This exercise revealed several discussion points in the evaluation of an NE component itself.

Current topics of work are language extensions, improvement of precision and recognition heuristics, and coreference questions.

[2]Details on CEP (= Content Extraction Pipeline) are given in the MULTISENSOR Deliverable D7.2, Technical requirements and architecture design.

2.1 Work progress in NER task

As the MULTISENSOR project progresses, the need increases to have a significant-size repository of SIMMOs (Socially Interconnected and Multimedia-Enriched Object), to be used as a training basis for the larger-scale extraction tasks. For this reason, a production chain was installed to populate this repository. The LT NER component is part of this production chain.

2.1.1 Interface adaptations

In the versions so far, there was no restriction on the input text. It was taken from URLs, normalised and tokenised, and offered as a starting point for the following components. In the new CEP setup, the text comes from a repository created by a crawling component. This component delivers not just the text in a ‚canonical' form, which prevents the NER from changing it (e.g. by a normalisation process); it also provides additional text specific annotations which the NER component must use, most notably {language} and {document-url}. This required an adaptation in the input interface.

2.1.2 Throughput

As it turned out in the tests that the services are heavily used (> 30 accesses per second), the system was restructured to cope with intensive throughput by supporting parallel processing.

2.1.3 Lexicon: Domain-tuning of lexical resources

Domain adaptation intends to optimise the NE recognition results, as it supports the general recognition strategy and underlying resources by domain-specific expertise. This holds in particular for lexical resources.

The lexical resources of the NER therefore were adapted to the three domains relevant for MULTISENSOR, which, for short, will be called ‚Energy' (energy policy, green energy etc.), ‚Household' (household appliances, descriptions, tests etc.), and ‚Dairy' (activities in the dairy market). Tuning consists of collecting and lexicalising entities from these three domains.

In order to keep track of the changes, and to keep the lexical resource maintainable, the lexical representation was changed by adding a feature ‚domain' (values being ‚energy', ‚household' and ‚dairy') to all domain-specific entries. This enables the recogniser to be made sensitive for the domain, in case e.g. a topic detection component is added to the processing chain.

Lexical resources for Energy:

Several resources were added to the lexicon for the energy domain:

- from Wikipedia, lists of nuclear power plants were added, POS being ‚*location*', subtype ‚*nucPP*', about 250 entries of source ‚*Wikipedia*';
- similar lists of offshore wind parks were added to the lexicon, POS being ‚*location*', subtype ‚*windoffsh*', about 120 entries of source ‚*Wikipedia*';

- fromReegle Energy Actors (www.reegle.info), companies and institutions were collected which are considered to be players in the energy domain. They are categorised as POS ‚*company*' or ‚*institution*', subtype ‚*energyactor*', about 1800 entities, however many of them with several allomorphs (like abbreviations: ‚*Marine Current Turbines*' as well as ‚*MCT*'), all of source ‚Reegle';
- from the MULTISENSOR repository, a number of articles was selected randomly, a test set was put aside, and a subset of the remaining texts was analysed for named entities. This activity resulted in about 1300 entries of different types, of source ‚*MS-Korpus*';

These entries were normalised and prepared for lexicon import by providing required annotations (POS, subtype, entry type, domain, allomorphs etc.).

Lexical Resources for Household:

For the Household appliances use case, no resources similar to the Wikipedia entries for energy could be found. The only resource to be used was the MULTISENSOR repository. Again, about 200 documents of the resource were randomly collected, and, after setting aside the test set, inspected for lexicalised named entity resources. About 500 entities were collected, mainly companies and brands, and annotated with POS, entry type, domain being ‚*household*', source being ‚*MS-Korpus*'.

Lexical Resources for Dairy:

The same situation for this use case was found as for the Household one: The only resource to be used was the MULTISENSOR repository. Like there, about 200 documents of the resource were randomly collected, and, after setting aside the test set, inspected for lexicalised named entity resources. About 1200 entities were collected, mainly institutions, companies and brands, and annotated with POS, entry type, domain being ‚*dairy*', source being ‚*MS-Korpus*'.

The lexicons of the three use cases were merged into one, and added to the overall lexicon repository. Conflicts are solved by keeping the more general reading (e.g. ‚*London*' is not only a 'household' term)

2.1.4 Grammar: Extension of recognition coverage

Previous versions of the NER were largely based on lexical information; i.e. they could recognise NEs when they were found in the lexicon. Of course, this is only a partial approach, and does not sufficiently consider cases where an entity (e.g. a person name) is not (yet) in the lexicon[3].

Identifying hitherto unknown entities in a text, and assigning the right type (‚*person*', ‚*location*' etc.) is the task of the recognition grammar of the LT-NER. The grammar uses contextual clues for such NE recognition.

[3] Adding a name to the lexicon, however, is always an option, and improves NER accuracy up to a desired level.

Grammar formalism:

The recognition formalism operates on linguistic objects described as feature structures. It manipulates these feature structures by using probabilistic context-free grammars. The approach is based on parsing proposals like Charniak (1997)[4] but extended by an elaborate feature handling mechanism and organised as rules with a phrase structure backbone and a feature manipulation section, whereby feature manipulation is done via operators which allow grammar writers to test existence and values of features, to unify, set and remove features and values, and manipulate tree structures.

Analysis strategy:

Terminal symbol

As no full analysis of a sentence is intended at NER level, a sentence (‚S') is seen as a (flat) sequence of textual (‚TxT') and NamedEntity (‚NE') nodes, which are recursively added to a left anchor:

S -> TXT *S -> S TXT*

S -> NE *S -> S NE*

At the end of the process, text nodes are skipped, and all NE nodes dominated by the S node are harvested.

NE Node

Each NE node needs to have at least the following features:

onset	start position in the input string
offset	end position in the input string
lemma	canonical form of the entity
textform	form of the entity as appears in the text (e.g. an abbreviated form)
netype	type of the entity; types are ‚*person*',‘*location*',‘*organisation*',‘*brand*',‘*time*', ‘*amount*'
nesubtype	subtype of the type

All other features are optional. The NE nodes are built by sets of rules, each covering a given type. I.e. there are rules for locations, persons, etc. Examples are given below.

Terminal nodes

Terminal nodes are created in different ways:

- Elements found in the **lexicon** are marked as ‚src=lex'. Lexical entries consisting of multiwords are composed by special grammar rules.

[4]Charniak, E. (1997): Statistical Parsing with a Context-free Grammar and Word Statistics. Proc AAAI. A justification of the approach is given in the MULTISENSOR Deliverable 2.2: Basic techniques for speech recognition, text analysis and concept detection.

- Elements not found in the lexicon are marked as **unknown**. Unknown tokens undergo a suffix string check, as sometimes the suffix of a word can be used as an indicator of its type (e.g. ‚Xxxitzky' -> ‚person', ‚Xxxtown' -> ‚location'). If such suffixes are identified the token is marked as ‚src=sufx', otherwise it is marked as ‚src=unk'
- Every token is also checked if it occurs in the **referents**' list which contains previously recognised entities which could serve as a referent for the current token. If it is found there, the system assumes that there is a possibility of coreference, and adds the reading of the referents' list , with ‚src=ref' annotation.

Multiple readings

The formalism is non-deterministic and is able to produce all possible readings of a given input chain. Therefore special attention has to be paid to compute the best possible reading, and deliver it first. This is done by using a scoring mechanism: Each rule gets a score, and the score a subtree is computed of the score of the rule and of the nodes involved.

As no tree banks for the intended kind of rules exist, rule scores were estimated by the rule writer, and balanced against each other in the testing process.

Analysis rules:

For each netype, a set of rules is written. The current version supports persons, amounts, and times.

Persons

The identification of persons uses the following lexical material:

first names	(NoPF). The lexicon contains a large set of possible first names
last names	(NoPP). These names are collected e.g. from ‚*the 100 most frequent French / English / German / … names*'
full names	(NoPs), like ‚*Alexander Pushkin*'. These names are complete and need not undergo any grammar treatment.
person indicators	(NoPPI). These are lexical elements indicating that a unknown token in their context could be a person name (like ‚*Professor Xxxx*'; ‚*his mother Xxxx*'; ‚*Xxxx explained*'). Indicators can be found before or after the unknown token.

First names and name indicators can convert names of other types into person names (‚*Washington*' -> location but ‚*George Washington*' -> person; ‚*Siemens*' –> organisation but ‚*Peter von Siemens*' -> person). Then the rule set for persons is roughly:

```
NE-Person      ->NoPs                                  // full name from lexicon
NE-Person      ->Pers-FirstName                        // only first name found in the text
NE-Person      ->Pers-LastName                         // no indicator! only for lexicalised names
NE-Person      ->Pers-FirstName+  Pers-LastName
NE-Person      ->Pers-Indicator (Pers-FirstName)* Pers-LastName
Pers-LastName  -><lexicalised> | <unknown> | <referenced> | <converted> (e.g. from NoPL)
```

Times and Amounts

Similar rules are provided for times (dates, hours), and amounts. The rules for these elements contain lexical elements (time tokens (NoPT) like ‚*Friday*', ‚*Christmas*', ‚*quarter*'; amount tokens (NoPU) like ‚*km*', ‚*$*', ‚*EUR*'). Combinations of such elements with numeric elements are collected by the grammar rules for these entities.

Organisations, Locations

Rules for these entity types are under development. While organisations need extended reference to indicators (like legal forms for companies: ‚*ABB*'), locations need more lexicalised approaches and tend to decrease precision when identified by contexts.

2.2 Evaluation

Test of the NER quality is essential for good results in the overall MULTISENSOR analysis chain. As NER have special requirements, test cases were prepared in addition to the MULTISENSOR test suites.

2.2.1 Test data

From the MULTISENSOR repository, test sets with MULTISENSOR documents were extracted: For each language, and each use case, about 200 documents were downloaded; of these, 20 documents were randomly selected as test documents. The documents were not further cleaned or processed; they contain formatting, spelling, and other errors. This resulted in 60 documents per language, 20 per use case. Tests were done for English, in the reporting period; other languages will follow. To be able to run evaluations, three documents were manually annotated, in a kind of ‚gold standard.

2.2.2 Test systems

To measure the progress between versions, the previous version (V1506) and the current version (V1510) were compared; each of them was tested in two flavours, one lexicon-based (V1506L / V1510L), the other one grammar-based, taking also contexts into account (V1506G / V1510G), resulting in four test systems:

V1506L	version 1, basic lexicon, only lexicalised NEs recognised
V1506G	version 1, also contexual indicators for NEs are used
V1510L	version 2, domain-adapted lexicon, only lexicalised NEs recognised
V1510G	version 2, domain-adapted lexicon, also contexual indicators for NEs are used

2.2.3 Evaluation results

Domain	Energy		Household		Dairy		Total	
System	recall	precision	recall	precision	recall	precision	recall	precision
V1506L	0.31	0.58	0.18	0.36	0.29	0.48	0.28	0.51
V1506G	0.37	0.45	0.20	0.27	0.38	0.43	0.33	0.40
V1510L2	0.71	0.76	0.68	0.69	0.69	0.73	0.70	0.74
V1510G2	0.74	0.79	0.70	0.69	0.71	0.74	0.73	0.75

Table 1: Evaluation of the system versions by domain

The result is given in Table 1. The following observations can be made:

- The second version (V1510) has significantly better results in both lexicalised (V1510L) and contextual/grammar (V1510G) versions than their first version counterparts. In terms of expectations, it is much better than the lowest expectation (which was 5% improvement for recall / precision), and even above the highest expectation (10%).
- Domain adaptation on the lexical level (i.e. lexicalising domain-specific named entities) is the biggest contribution to recognition quality in the test documents, with an increase of 0.4 in recall and precision. Contextual heuristics contribute in a rather moderate way; however, the V1 grammar (V1510G) is still under development, so improvements can still be expected
- The grammar in the first version improves recall but lowers precision vis-à-vis the lexicalised version. I.e. the contextual heuristics provided by the grammar introduce a significant amount of noise. This is not the case in the second version any more, where the grammar-based contextual analysis improves recall but also improves precision. This is due to refined contextual analysis.
- As for the different domains, it can be seen that the household documents are the most difficult to analyse; this may be due to the fact that the documents of this domain show a significantly lower input quality than the other domains.

If only the entity 'person names' is considered (see Table 2), it can be seen that contextual analysis contributes to recall and precision more than in other NE types; however, overall precision has room for improvement. An error analysis shows that a major source for introduction of noise are homographs between common nouns which are capitalised in headings and proper names (e.g. '*Kitchen'*, '*List*', '*Glass*', '*Handy*' etc.); actions are required to give such homographs a special treatment, independent of other person names.

	Energy		Household		Dairy		Total	
	recall	**precision**	**recall**	**precision**	**recall**	**precision**	**recall**	**precision**
V1506L	0.15	0.08	0.64	0.20	0.16	0.08	0.29	0.13
V1506G	0.57	0.30	0.76	0.40	0.52	0.33	0.61	0.33
V1510L2	0.47	0.34	0.71	0.31	0.49	0.40	0.54	0.34
V1510G2	0.79	0.50	0.85	0.35	0.69	0.53	0.78	0.45

Table 2: Evaluation of NE-type 'Person'

Overall, the following steps will be taken:

- Improvement of the existing NE grammar, completion of coverage, refinement of contextual heuristics; the target could be a ratio of 0.8 for both recall and precision
- Extension to other languages; first versions of Spanish, German, and French are underway. These languages require a change in the contextual analysis (e.g. capitalisation is not a good NE indicator in German)

3 CONCEPT EXTRACTION FROM TEXT

3.1 Work progress in concept extraction task

Concept extraction is addressed by compiling a list of terms by analyzing corpora of documents belonging to each use case and then using this list to detect mentions of the concepts in texts. Three different subtasks are foreseen: term extraction, mention detection and word sense disambiguation. Specific tools and resources are used for each task.

3.1.1 Term extraction

The goal of term extraction is to obtain a list of candidate terms denoting concepts relevant for the domain at hand. This list of words and multiword expressions can be ranked according to a termhood metric obtained through the statistical analysis of a corpus of domain documents. For the concept extraction service developed as part of the MULTISENSOR project we have used TermRaider[5], a term extraction tool released as a plug-in of the GATE framework.

TermRaider identifies term candidates in the documents and scores them using one of the metrics supported by the tool. After processing a test corpus using TermRaider and doing a preliminary evaluation, we determined Kyoto domain relevance score (Bosma and Vossen, 2010) as the best suited for the task. Kyoto score is calculated according to the formula $d_f * (1 + n_h)$, where d_f is the document frequency of a term candidate and n_h is the number of its distinct hyponymous term candidates found in the corpus.

3.1.2 Mention detection

Mention detection operates on the texts analyzed by the content extraction pipeline. The ultimate goal of the concept extraction service is to produce annotations of references to concepts disambiguated against a reference body of knowledge. We have chosen BabelNet[6] as our resource of reference, due to its lexicographic and encyclopaedic coverage of multilingual terms resulting from a mapping of Wikipedia pages and WordNetsysnsets. The version integrated in the MULTISENSOR system is 2.5.1, which is the latest version of the database for which dumps are available for download.

The concept extraction service detects candidate mentions by detecting nominal groups in the text using the OpenNLP[7] chunker and looking them up in the BabelNet database. Since BabelNet indexes concepts using the list of redirections from Wikipedia and the lists of synonyms from WordNet, the look-up will cover many equivalent expressions used to refer to the same concept (a BabelNet index).

Rather than marking as concepts all text fragments that are found as indices in BabelNet, the concept extraction service only annotates candidate mentions which are considered

[5]https://gate.ac.uk/projects/arcomem/TermRaider.html

[6]http://babelnet.org/

[7] https://opennlp.apache.org/

potentially relevant to the domain and have not been annotated as NEs by the NER service. The mentions annotated by the service therefore exclude references to general concepts which aren't very informative for the use case. The mechanism that filters candidate mentions is based on the terminology list produced the term extraction task. A fixed threshold is used on the lists produced by TermRaider for each use case, so that only top-scored term candidates which are also indexed by BabelNet are marked in the text by the concept extraction service.

3.1.3 Disambiguation of concepts

While some mentions lead to a single BabelNetsynset, many others index multiple concepts. These polysemous mentions require disambiguation against BabelNet. A word sense disambiguation tool for BabelNet, Babelfy[8], was released a year ago but is only available as a RESTful service with a limit on the number of calls that can be made to it. Given that one of the requirements of the MULTISENSOR system is that a large number of documents are analyzed, we chose to implement our own disambiguation tool for the BabelNet database.

Like Babelfy, our disambiguation strategy is graph-based and aims at finding a global, coherent disambiguation of mentions in a document, so that the disambiguation of each mention is conditioned by the disambiguation of other mentions in the document. In a first stage we are replicating the Babelfy approach to disambiguation, which we plan on adding to the MULTISENSOR system shortly after the publication of this deliverable. In the following months we will also attempt to improve the approach. The improvements will be based, on the one hand, on weighting the relations between BabelNetsynsets according to their type and effectiveness in disambiguating mentions of use case entities. On the other, we will go beyond the Babelfy disambiguation metrics which are based exclusively on the BabelNet semantic graph, and will incorporate distance metrics between mentions in the text obtained through vectorial semantics.

3.1.4 RDF modelling of concept annotations

The concept extraction service produces semantic annotations encoded as RDF triples and serialized using JSON-LD. For this purpose the NIF model is used to model stand-off character-based annotations in a similar fashion to the annotations produced by the NER service. Unlike annotations of mentions to NEs, there is no reference to a NERD entity type for concepts. Instead, only a reference to the BabelNetsynset URI is produced using the ITS 2.0[9] *taIdentRef* property.

3.2 Evaluation

The evaluation of the concept extraction service is based on a gold standard manually annotated with mentions to concepts. The performance of the service is measured using

[8] http://babelfy.org/

[9] http://www.w3.org/TR/its20/

precision, recall and f-score figures obtained by comparing the annotations produced by the service against the gold standard.

3.2.1 Annotation of a gold standard

One of the main hurdles in evaluating concept or term extraction services is that a gold standard corpus annotated with those concepts is required. For tools geared toward specific domains, as is the case of the MULTISENSOR concept extraction service, the corpus must contain only annotations of concepts which are relevant to the domain at hand. While at least one corpus annotated with BabelNetsynsets is available[10], and others have been released for other large datasets[11][12], none of them fit our requirements, namely (i) it has been manually annotated or revised so that it can be used as a gold standard, (ii) it is annotated with the concepts in our reference body of knowledge (BabelNet), and (iii) identifies the concepts relevant to the MULTISENSOR use cases.

For this reason we decided to create a set of annotation guidelines and proceed with the annotation of a set of 60 sentences obtained at random from the use case corpora, 20 sentences for each. The guidelines (see Appendix A) give criteria for the manual annotation of NEs, domain concepts, coreference links, and FrameNet or VerbNet n-ary relations. The 60 sentences are being annotated separately by three annotators using a version of the Brat tool[13] adapted to support normalization against BabelNet and FrameNet. A consensus annotation is also being created from their annotations. This consensus annotation will constitute the gold standard to be used in the evaluation of the concept extraction and relation extraction services[14]. At the moment **the gold standard consists of 10 sentences belonging to the energy policies use case.**
The three annotators agreed in 75% of the concept annotations in average, and in 93% of the Babelnet synsets assigned to them. We do not provide inter-annotator agreement measures which account for chance agreement such as Cohen's Kappa due to the complexity of determining the probabilities of chance agreement. Future deliverables will address these complexities and provide such metrics.

3.2.2 Results

We processed the 10 sentences of the gold standard with the concept extraction service and a baseline consisting of a chunker where all maximum spanning nominal chunks were marked as concepts. A two-fold evaluation of the output was conducted where the baseline and system annotations were first evaluated against all concepts annotated in the gold standard, which include not only domain-specific (related to energy policies) concepts but also any concepts that the annotators considered relevant to the meaning of the sentences.

[10] http://lcl.uniroma1.it/MASC-NEWS/

[11] http://googleresearch.blogspot.com.es/2013/07/11-billion-clues-in-800-million.html

[12] http://datahub.io/dataset/dbpedia-spotlight-nif-ner-corpus

[13] http://brat.nlplab.org/

[14] http://brat.taln.upf.edu/#/goldstandard/EN/consensus/

The same baseline and system outputs ere then evaluated by comparing only against concepts marked as strictly terminological in the gold standard.

The results are shown in Table 3 in terms of precision and recall. The values show that the baseline outperforms the basic version of the concept extraction service in precision and particularly in recall. **This is below the lowest expectation set in D1.1 (increase of 10% in both metrics when compared to baseline) and indicates that there is much work to do until the performance of the relation extraction service reaches reasonable performance**. Besides, the lack of a proper disambiguation strategy meant that an evaluation of the references assigned to concepts was not possible, but remains part of our future plans.

	Baseline		**Concept extraction**	
	All concepts	**Terminological concepts**	**All concepts**	**Terminological concepts**
Precision	0.59	0.13	0.38	0.13
Recall	0.69	0.46	0.075	0.025

Table 3: Results of the evaluation of concept extraction

4 CONCEPT LINKING AND RELATIONS

4.1 Work progress in concept linking and relations task

The extraction of relations focuses on identifying coreference links between text fragments and relations between concepts and NEs indicated by linguistic predicates. Coreference links are established between explicit mentions annotated in the previous steps and anaphoric or cataphoric expressions such as pronouns, under the assumption that both denote the same entity or concept. Conceptual relations between concepts and NEs are identified by analyzing predicates like verbs and predicative nouns, and their functional arguments. A deep dependency parser annotates a text with syntactic structures that mark linguistic predicates and their arguments. Relations are then extracted from this structure by classifying the predicates according to a repository of predicative senses, assigning semantic roles to their arguments and determining when an argument corresponds to a previously annotated concept or NE. The resulting relations are n-ary relations and involve concepts, NEs or other relations.

4.1.1 Dependency parsing

Dependency parsing constitutes the first step in the extraction of relations from the text. State-of-the-art syntactic dependency parsing delivers surface-syntactic structures (SSyntSs), which are per force idiosyncratic in that they are defined over the entire vocabulary of a language (including governed prepositions, determiners, support verb constructions, etc.) and language-specific grammatical functions such as, e.g., SBJ, OBJ, PRD, PMOD, etc.; see, among others (McDonald et al., 2005; Nivre et al., 2007; Bohnet and Kuhn, 2012; Dyer et al. 2015). On the other hand, semantic (or deep) parsing delivers logical forms (LFs) or semantic structures (SemSs) equivalent to LFs, PropBank (Kingsbury and Palmer, 2002) or FrameNet (Fillmore et al., 2002).

For NLP-applications such as in MULTISENSOR, which aim at not only filling databases but also perform text-generation in a posterior step, neither SSyntSs nor LFs or SemSs are adequate: the high idiosyncrasy of SSyntSs is obstructive because of the lack of semantic information, while current LFs and SemSs are problematic mainly because of the loss of syntactic information or meaningful content elements and/or relations between them. Besides, multiword expressions which have been marked as denoting a specific entity by the NER and concept extraction services constitute atomic units of meaning that require no analysis of its parts. For this reason in the input to our parser all multiword expressions are marked as a single token.

“Syntactico-semantic” structures in the sense of deep-syntactic structures (DSyntSs) as defined in the Meaning-Text Theory (Mel’čuk, 1988) are in this sense arguably more appropriate. DSyntSs are situated between SSyntSs and LFs/SemSs. Compared to SSyntSs, they have the advantage to abstract from language-specific grammatical idiosyncrasies. Compared to LFs, PropBank and Frame structures, they have the advantage to be complete, i.e., capture all and distinguish all argumentative, attributive and coordinative dependencies between the meaning-bearing lexical items of a sentence, and to be connected. In other words, DSyntSs allow for a more straightforward mapping to abstract ontological structures

which the pipeline, of which WP2 is responsible for. In this document, we report on the work we have been carrying out on English and Spanish.

In Deliverable D2.2 (Section 5.3), we give a general overview of deep-syntactic parsing. In this section, we give the details about the system and the resources used.

In order to get the deep-syntactic structures, we first trained (Bohnet and Nivre, 2012) transition-based parser, which combines PoS tagging, morphosyntactic tagging, lemmatization and syntactic labeled dependency parsing. The SSyntS parser was trained in 25 training iterations, using in each iteration the model from the preceding iteration for further processing. Once we have a predicted SSyntS, we run the deep-syntactic parser developed in the framework of MULTISENSOR (described in (Ballesteros et al., 2015)) which is a cascade of three SVM classifiers:

1. Hypernode identification: For hypernode identification, we trained a binary SVM with polynomial kernel from LIBSVM (Chang and Lin, 2011). The SVM allows for both features that are related to the processed node and higher-order features, which can be related to the governor node of the processed node or to its sibling nodes; and it basically classifies the nodes to be kept in the DSynt structure or to be removed.

2. Tree reconstruction: The tree reconstruction algorithm find the best head for the nodes that do not have a head after removing some nodes in **1.** The output of this step is a well-formed DSynt tree with SSynt labels.

3. DSynt arc labelling: Once we have the tree from **2.**, the next step is to find the correct deep-syntactic labels for each edge/arc in the tree. This is obtained by training a multiclass SVM with polynomial kernel from LIBSVM. The SVM uses surface features of the two nodes that have some participation in the edge plus contextual features.

The deep-syntactic parser has been trained on parallel SSyntS and DSyntS corpora. For English, we use the dependency version of Penn Treebank 3 (Johansson and Nugues, 2007). We derived from it the deep-syntactic (DSynt) annotation –aligned with the SSynt annotation- in which we removed all determiners, auxiliaries, *that* complementizers, infinitive markers to, punctuations and functional prepositions of verbs and predicative nouns. In order to obtain a DSynt annotation of relations with a high quality, we used existing manually annotated lexical resources during the derivation, namely, PropBank (Kingsbury and Palmer, 2002) and NomBank (Meyers et al., 2004). For Spanish, we use the manually annotated AnCora-UPF SSyntS and DSyntStreebanks (Mille et al., 2013), which we adjusted for our needs. The output of the deep-syntactic parser is a connected tree with only meaningful nodes, attribute-values associated to each node (for storing the tense, voice, semantic number, etc.), and MTT's deep-syntactic relations (*I*, *II*, *III*, *IV*,*V*, *VI*, *COORD*, *ATTR*, *APPEND*, *NAME*) with a direct mapping to the PropBank/NomBank terminology for English, as shown in the following Figure 4.

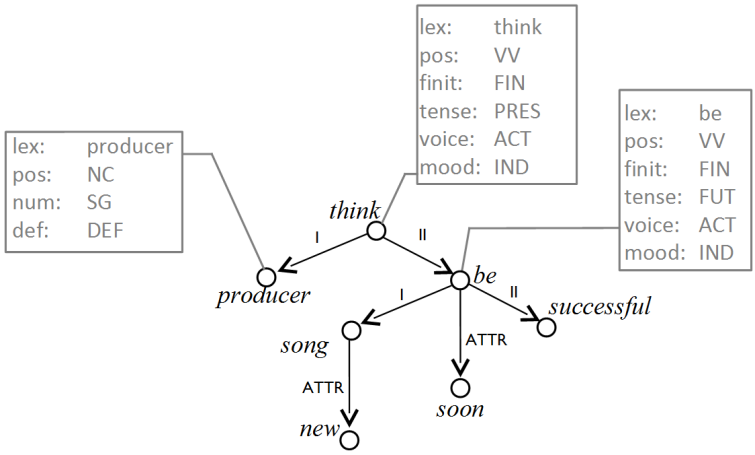

Figure 4: A sample deep-syntactic structure for the sentence *The producer thinks that the new song will be successful soon*[15]

4.1.2 Coreference resolution

Coreference resolution refers to finding text fragments which refer to the same entity. While NER and concept extraction are able to find equivalent expressions used to refer to the same entity, they are not able of detecting abbreviated forms that refer to discourse entities introduced elsewhere in the text. Examples of abbreviated forms include repetition of nominal constructions where modifiers are dropped in latter mentions, and anaphoric expressions involving pronouns or other proforms.

For the MULTISENSOR pipeline we decided to use the coreference resolution module of the Stanford CoreNLP tools[16]. This deterministic module has been adapted to recognize the annotations of entities and concepts produced by the NER and concept extraction service as potential coreferring mentions. The output of this module is a set of coreference chains which group together expressions which refer to the same entity.

4.1.3 Disambiguation of predicates and semantic role labelling

The extraction of relations from the structures produced by the dependency parser involves determining the relation or situation that a linguistic predicate signifies in a given context, and also finding its participants from the syntactic arguments of the predicate. For instance, the predicate “elect” can be used to refer to the act of electing a representative in an election. This situation involves at least three participants: (i) the people who elect the representative, (ii) the entity that is elected, and (iii) the position for which the representative is elected. In order to identify this type of situations in a text, we need to first disambiguate the predicate “elect” and see if it refers to this situation or is used in some other sense, and second to find the participants in this situation.

[15] *ATTR* maps to *AM*, *I* to *A1*, *II* to *A2*, *III* to *A3*, etc. If the predicate has external arguments, all PropBank and NomBank argument numbers start from *A0* instead of *A1*.

[16] http://nlp.stanford.edu/software/corenlp.shtml

For these tasks we use the **FrameNet** (Fillmore et al., 2002) repository of predicative senses. FrameNet defines frames which specify a set of roles (frame elements) which must be filled by participants (core arguments) as well as optional roles taken by additional participants (non-core arguments). Language-specific dictionaries associate linguistic predicates (lexical units) to frames. We disambiguate linguistic predicates found by the deep parser against the frames they have assigned in FrameNet, and then go through the arguments of the predicate and assign to them the roles specified by the frame. For instance, in the case of "elect" the corresponding frame is "Change_of_leadership" which has three core frame elements "Selector", "Body", and "Function".

In prevision of the pipeline described for WP6, in which generation is performed starting from abstract structures that do not contain any linguistic information, we also intend to link the concepts to VerbNet and PropBank/NomBank entries, and the participants to VerbNet and PropBank/NomBank roles[17].

The **VerbNet** classes (Schuler, 2005) are different from the FrameNet frames in that predicates are grouped according to not only semantic criteria, but also syntactic ones. For instance, the frame "Change_of_leadership" has two possible mappings to VerbNet classes, namely 10.1, a class named *remove*, which contains verbs such as "abolish", "deduct", "depose", etc., and 29.1, a class named *appoint*, which contains verbs such as "appoint", "elect", "install", etc. The VerbNet classes and roles are more generic than the FrameNet ones and facilitate the mapping towards more surface-oriented structures (see WP6); there are currently 270 different classes and 38 different roles (such as *Agent*, *Beneficiary*, *Goal*, etc.). Through SemLink (Palmer, 2009), the mapping to FrameNet frames is established for all the classes.

FrameNet is not the only resource which can be used to determine the sense of predicates and assign roles to their arguments. The **PropBank** (Kingsbury and Palmer, 2002) and **NomBank** (Meyers et al., 2004) resources are not based on classes of words. Rather, the entries are lexical units, as, e.g., *elect.01*, *elect.02*, for which the list of arguments is listed (first argument, second argument, etc.). Most lexical units assigned a **VerbNet** class and/or FrameNet frame, and their arguments are assigned VerbNet roles. VerbNet classes and argument roles can be used as an alternative (to FrameNet) repository of predicative senses. The PropBank and NomBank argument numbers correspond to the deep-syntactic relations we obtain in our parsing pipeline, so these annotations are compatible. PropBank and NomBank contain in total the description of nearly 12,000 lexical units, of which 43% have a mapping to VerbNet classes.

In the following two subsections we describe two alternative methods to identify FrameNet frames or VerbNet classes and their participants.

[17]Disambiguating against VerbNet and Propbank/NomBank can also serve for a better mapping to FrameNet Structures.

4.1.4 Semafor

In this section we describe how Semafor (Das et al., 2010) can be used to assign FrameNet frames and frame elements to predicates and their arguments. Semafor works with texts in English and produces annotations corresponding to frames and their elements. Internally, it uses a surface dependency parser and two log linear models trained on Framenet 1.5, one model to assign frames to predicates and another to identify the frame fillers (arguments) for each one of the frames.

For the integration of the system in the MULTISENSOR pipeline, we introduced some modifications to the Semafor tool. First, we changed it to accept as input the CoNLL'09 files produced by our dependency parser (Semafor uses by default its own dependency parser). The original Semafor tool marks frame element fillers as text spans (see Figure 5). We aim at assigning concepts or NEs as frame element fillers, when possible. For this reason we modified Semafor to look for the head of the span and annotate it with the frame element. In those cases where the head corresponds to a word or multiword expression annotated with a reference to a concept or NE, that entity becomes the participant in the frame.

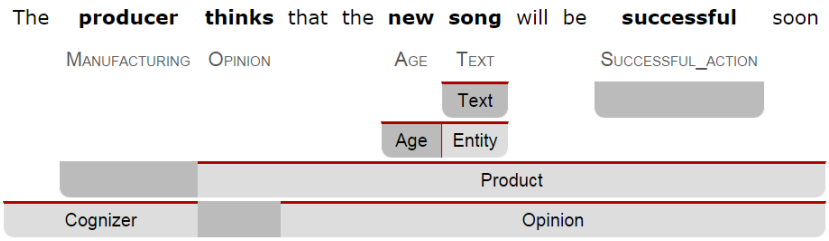

Figure 5: A sample Semafor output

We faced two problems when integrating Semafor into the MULTISENSOR pipeline:

- The Semafor library is not thread safe and therefore it is not possible to run it concurrently. This is a major hurdle for the population of the MULTISENSOR semantic repository with information extracted from large amounts of documents.
- Semafor does not give syntactic information about the components of the sentence (government patterns, subcategorization information, etc.) needed for the generation of abstractive summaries.

As a consequence, we decided to implement another frame-semantics parser which does not suffer from these problems.

4.1.5 Rule-based method

For ontology-to-text generation (WP6) we need a description in syntactic terms of the lexical realizations of FrameNet/VerbNet relations extracted from texts. A mechanism to achieve this is to associate FrameNet frames and VerbNet classes to their lexical realizations in lexical databases of predicative words. PropBank and NomBank are two such databases for verbs and predicative nouns, respectively. Their entries contain, for each word sense, a description of its predicate-argument structure (in other words, its deep-syntactic structure). SemLink (Palmer, 2009) is a resource that maps PropBank and NomBank predicate entries and their

arguments to VerbNet classes and roles respectively. Additionally, SemLink also contains a mapping from VerbNet classes and roles to FrameNet frames and frame fillers.

We have used SemLink to implement a module for the annotation of deep syntactic (predicate-argument) structures produced by our parser with syntactic properties, VerbNet classes and roles, and FrameNet frames and frame elements. This module, implemented with graph-transduction grammars in the MATE environment (Bohnet and Wanner, 2010), replaces Semafor. The annotation is performed in two steps: first we convert the deep-syntactic structure onto a VerbNet structure, and then the VerbNet structure onto the FrameNet structure. Since VerbNet does not cover all predicative word types but only verbal and nominal predicates, we have extended the resource to include adjectives, semantic adverbs, semantic prepositions and conjunctions. A sample output of the current module is given in Figure 6.

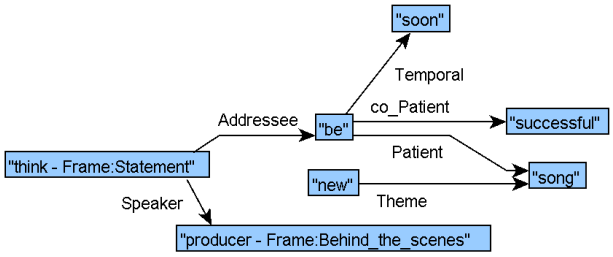

Figure 6: A sample output of the rule-based frame-semantics parsing

In the first version of this new module:

- As in the case of Semafor, only English is supported.
- the predicates are not disambiguated at any level, that is, if there are more than one possible mapping:
  - for each word, we select the first (most common) meaning in PropBank/NomBank
  - for each VerbNet class and FrameNet frame we select the first one according to the alphabetical ordering (this is why the incorrect frame and frame fillersare assigned to “think” in Figure 6).
- if no mapping is found for a predicate/class, we leave the annotation of the previous layer in place; for instance, if neither FrameNet frame nor VerbNet class is assigned for NomBank entry “success.01”, we keep “success.01” as the label for the predicate in the output.
- if no mapping is found for an argument/role, we use as default mapping the most common label.

In the next version of this module, we intend to perform sense disambiguation according to PropBank/NomBank, VerbNet and FrameNet and make them accessible during the graph-transduction pipeline, so as to improve the MATE mappings. We will also adapt this pipeline to other languages covered in MULTISENSOR.

4.1.6 RDF modelling of relation annotations

Modelling the annotations produced by the relation extraction service is a fairly complex task when compared to other services. We kept the NIF 2.0 core ontology as the model for

stand-off annotations. Given that the OLiA ontologies do not cover semantic annotations like those produced by the service, we had to look elsewhere. The FrameNet database is only available as a set of XML files. We chose a recent RDF conversion by Nuzzolese et al. (2011), and had to rework a model that integrates NIF and FrameNet RDF without introducing any custom properties or classes.

Figure 7 shows the model used to integrate FrameNet RDF and NIF, while Figure 8 shows a simplified view of how a sample sentence is assigned frame and frame elements.

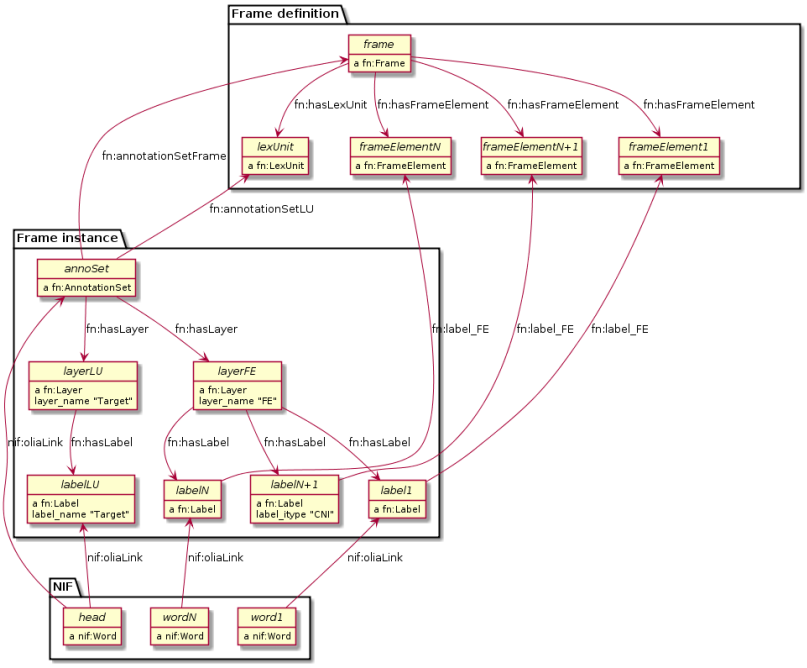

Figure 7: Integration of FrameNet and NIF

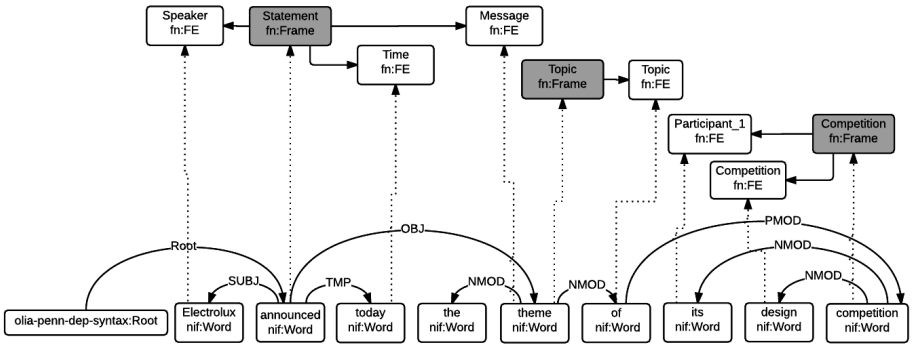

Figure 8: A sentence annotated with frames and frame elements

At the moment only FrameNet annotations are produced as RDF. In the following months we plan on investigating ways of modelling annotations based on the other resources covered by the SemLink project.

4.2 Evaluation

4.2.1 Evaluation of the dependency parsers

We carried out an evaluation of the dependency parsers on the evaluation set of the corpora used for training.

For the evaluation of the SSyntS parsers, we calculated (i) the unlabeled attachment score (UAS, percentage of predicted tokens with a correctly predicted governor), and (ii) the labeled attachment score (LAS, percentage of predicted tokens with a correctly predicted governor and correct label).

	POS	LEMMA	LAS	UAS
Spanish	96.05	92.10	81.45	88.09
English	97.50	99.46	89.70	92.21

Table 4: Results of the SSynt parser evaluation

	Spanish	English
F1h	96.31	98.59
LAP	68.31	81.70
LAR	77.31	85.80
LAR	67.26	82.05
UAR	79.22	86.17

Table 5: Results of the DSynt parser evaluation

For the deep-syntactic parsers, the evaluation was carried out on two aspects:

- Hypernode identification evaluation:
  - F1h (F1h) = 2ph*rh /(ph+rh), being ph the number of correctly predicted nodes divided by the total number of predicted hypernodes, and being rh the number of correctly predicted hypernodes divided by the number of hypernodes in the gold standard.
- Dependency labels evaluation:
  - Unlabeled attachment precision (UAP): number of nodes with a correctly predicted governor divided by the total number of predicted nodes.
  - Labeled attachment precision (LAP): number of nodes with a correctly predicted governor and governing relation label divided by the total number of predicted nodes.
  - Unlabeled attachment recall (UAR): number of nodes with a correctly predicted governor divided by the total number of gold nodes.

- Labeled attachment recall (LAR): number of nodes with a correctly predicted governor and governing relation label divided by the total number of gold node.

We will provide the comparisons with the baselines and assess the performance according to the indicators in the final deliverable.

4.2.2 Annotation of a gold standard

For the evaluation of the relation extraction service, the same corpus of 60 sentences in English introduced in section 4.2.2 is being manually annotated with n-ary relations by three annotators using the Brat tool, and a consensus annotation is being created. The annotation of relations is also described in the guidelines (Appendix A). At the moment the gold standard of relations contains same subset of 10 sentences belonging to the energy policies use case and used for the evaluation of the concept extraction service. The three annotators agreed in 74% of the predicate annotations in average, 91% of the frames assigned to predicates and 85% of the annotated arguments. As in section 4.2.2, we do not provide inter-annotator agreement measures which account for chance agreement such as Cohen's Kappa due to the complexity of determining the probabilities of chance agreement for each subtask. Future deliverables will address these complexities and provide such metrics.

4.2.3 Results

	Semafor baseline			**Rule-based**		
	Predicates	**Frames**	**Arguments**	**Predicates**	**Frames**	**Arguments**
Precision	0.73	0.55	0.63	0.79	0.0	0.63
Recall	0.39 (0.57)	0.33 (0.5)	0.20	0.87	0.0	0.67

Table 6: evaluation results of the semafor and rule-based relation extraction pipelines.

Our analysis pipeline addresses three subtasks, namely the identification of predicates, the annotation of predicates with frames, and the identification of predicate arguments. We processed the 10 sentences of the gold standard with the Semafor-based pipeline, taken here as a baseline, and our own rule-based system.

The results are shown in Table 6 and show that our system is already not far to our highest expectations set in D1.1 (20% improvement in both precision and recall over baseline). Recall has been massively improved when compared to Semafor, even when excluding predicative constructions detected by our system but not considered by this tool (i.e. possessives, appositions, semantic prepositions and adverbs, conjunctions and other discourse markers). The recall figures for the baseline in Table 6 are given for all predicates marked in our gold standard and, in brackets, excluding predicative types not supported by Semafor. Recall in our system for predicate and argument detection increases from 17 to 38%, which is a notable improvement over the baseline.

Despite the promising results, due to lack of disambiguation our system is unable of assigning frame and role classes to predicates and their arguments respectively. This is in contrast to the Semafor baseline which does produce such annotations. The results of evaluating frame identification have been included in Table 6. Both precision and recall are 0 for our system, which assigns frames at random and did not manage to get a single one right in the 10 sentences used for the evaluation.

5 AUDIO TRANSCRIPTION

In D2.2 we gave a brief introduction into the state of the art in the field of speech recognition and described briefly our approach and the technical framework, as well as the data acquisition and data preprocessing tasks for language modelling. In the first project year the baseline system for English was released.

Since then the baseline system for German has been released and first steps towards the advanced versions have been made. Furthermore, we have started to manually transcribe many audio files in order to create test sets for both languages. We started with German and a test corpus large enough to start automatic tests of the component has been created. For English, we consider the option to “borrow” the test set from the partner project EUMSSI, since the manual transcription of audio data is a very expensive and time-consuming task.

In this deliverable we describe the work done during the second year of the project.

5.1 Work progress in audio transcription task

One of the main tasks during the reporting period was the release of the baseline ASR for German. The processes of data acquisition and preparation have already been described in D2.2.

The main new focus during the second year was on the post-processing of the ASR output in both languages English and German. The output of our ASR system, like of most other systems as well, is:

- in all-lowercase format
- with numbers and abbreviations represented in their fully spelled forms instead of their usual written representation
- without any recognisable sentence and phrase boundaries
- with subwords not yet joined into possible compounds (our approach is an open-vocabulary approach, where not only words, but also word fragments are trained and recognised respectively; s. D2.2)

Such a kind of text can be difficult to be analysed by the text analysis tools, thus the first steps towards more advanced versions have been to reconstruct the orthography, write numbers as symbols instead of words, create possible compound words from word fragments, and find plausible utterance boundaries.

5.1.1 Reconstruction of orthography

The recovery of capitalisation, technically also called “truecasing”, consists of assigning to each word of a text its corresponding case (lower, capitalised, all-upper, or mixed), and to recover the capitalisation of the first word in a sentence. For this, we basically apply two methods:

- The casing information can be inherent to the words as member of a certain part of speech (verbs are written in lower case, names capitalised), which areas such manageable through lexicons or word lists
- However, very often the correct capitalisation depends on the context. English words “house” and “common” are written lowercased if used in their generic noun or adjectival

meaning, but as parts of the named entity “House of Commons” both of them are capitalised. For this, other techniques are adopted, such as multiword lexicons, n-gram based statistical methods, and integration of results from the named entity recognition module.

Of particular difficulty is the nominalisation problem in German, having in mind that all nouns are written in capitalised form in German, and practically each word can be nominalised in some contexts. That is why in these cases the lexicon or the list lookup option cannot bring the expected results, but only taking into account of wider lexical and syntactical context.

5.1.2 Reconstruction of numbers

For the creation of the language models and recognition lexicons, all numbers, abbreviations, symbols, and dates had to be expanded from their written forms (2, etc., $, April 2nd) to their spoken forms (two, et cetera, dollar, the second of April). Only the word forms expanded in this way have been included into the recognition lexicon and can appear in the ASR output. Now, the job of a post-processing component is to recognise and change them back to their most usual written form. However, this process is not a 1:1 mirrored preprocessing part; it demands a slightly different approach in pattern recognition and replacement, such as:

- While in the preprocessing step all possible variants had to be mapped onto one normalized spelling variant (20 $, 20$, $20, 20 Dollar => “twenty dollar”), in the post-processing step decisions should be made how to represent a given word sequence in the respective symbolic way (“twenty dollar” => 20$ or $20?)
- In many cases, context should be taken into account (“paragraph twenty” can be changed to “§20”, but “in this paragraph” the word paragraph remains unchanged)
- The quality of the post-processing step also strongly depends on the quality of the recognition. For example, wrongly recognised word endings can impair a correct reconstruction of date patterns (e.g. German: “am drittenapril” is correct and can be replaced by “am 03. April” or “am 03.04.” or “am 03/04”, whereas “am drittemapril” is wrong and cannot be mapped to any correct date representation

For MULTISENSOR, both routines the pre-processing and the post-processing have been developed as Java programmes. As resources, they use language-dependent linguistic rules and dictionaries to identify, decompose or compose the respective patterns.

5.1.3 Finding utterance boundaries

The output of an ASR system is a long string of text, without any boundaries between utterances or phrases. As such it is very difficult for both human reading and machine processing. In MULTISENSOR, the speech recognition output goes into the text analysis pipeline and is supposed to be processed by the named entities recognition, dependency parsing, relation detection and other analysis tools in the same way as any other text input. But since such tools usually operate on sentence level or at least on sentence-like units, they are unable to deal with extremely long strings. For this, at least a chunking in smaller text portions is necessary.

The most desirable result would be sentences and phrases correctly detected and their boundaries separated by respective punctuation marks (period, question mark, and comma). But, an important aspect of the sentence boundaries detection is the nature of the spontaneous speech as such: people do not really speak in sentences. The natural speech flow rather consists of repetitions, hesitations, corrections, incomplete thoughts and statements, ellipses, interruptions and speakers talking over each other. Add to that the fact that these issues are difficult to be modelled and learned, they lead to even additional recognition errors. Experiments show that inserting punctuation marks on top of that, often leads to additional errors and makes the input for the subsequent analysis tasks less readable.

Instead we use a different approach by exploiting non-textual acoustic features. In addition to the pure textual recognition we also identify non-speech events (silence, hesitation, noise) in the audio signal and measure their length of time for each occurrence. To allow for a standardization of different speaking rates across different recordings we also calculate the minimum, average and median length of non-speech events for each audio file. This information is included in the audio transcription output and can be used by subsequent modules in the CEP pipeline.

5.1.4 Joining word fragments

For the training, big amounts of selected compound words were split into word fragments, modifiers were marked by plus signs (tischbein =>tisch+ bein), and the word fragments were included into the recognition lexicon. We have trained the fragment LM (language model) together with the full-words LM. For English and German we used two different approaches to this issue: German fragments are rather on the lexical level and contain lexicalised parts of compound words, while the English fragments are on the morphological level and contain prefixes, suffixes and word endings.

After the recognition, the fragment sequences have to be recovered and appropriate joined spellings generated. For this, we primarily rely on the LM quality and simply join all sequences of words ending in fragment markers with their subsequent words, except of cases with longer silences and pauses, recognised speaker turns or other segment changes between the fragments.

5.2 Evaluation

The transcription accuracy is usually measured by WER (word error rate). Given a reference transcription (ground truth: what was actually said) and the ASR output (hypothesis of what was said), the distance in words between them is calculated: how many words have to be inserted/deleted/substituted in the hypothesis to achieve the same accuracy as in the reference file?

The formula is:

$$\mathrm{WER} = \frac{100\ x\ (insertions + substitutions + deletions)}{total\ words\ in\ reference}$$

5.2.1 Evaluation tool

We use the open source tool NIST sclite[18] to calculate WER. It offers different evaluation outputs per test file for better error analysis:

1. It aligns hypotheses with references and marks insertions/deletions/substitutions:

```
Scores: (#C #S #D #I) 8 2 1 1
REF:  veranstaltet hat    das    rennen  der    ****   ADAC    BERLIN-BRANDENBURG und    die    IG     seifenkisten
HYP:  veranstaltet hat    das    rennen  der    ADFC   BERLIN  BRANDENBURG        und    die    **     seifenkisten
Eval:                                     I      S      S                                 D
```

2. It provides statistics on word pairs which are (frequently) being confused:

```
CONFUSION PAIRS                    Total                    (6)
                                   With >=  1 occurances    (6)

   1:     1  ->  die ==> sich
   2:     1  ->  hat ==> hatte
   3:     1  ->  heben ==> geben
   4:     1  ->  heben ==> hebeln
   5:     1  ->  nähe ==> näher
   6:     1  ->  produktfilm ==> produktfehlen
     -------
```

3. It gives overviews of error percentages by each speaker marked in the reference file:

interview-dialog.0003_speaker_7	5	52	65.4	19.2	15.4	5.8	40.4	80.0
interview-dialog.0003_speaker_8	1	3	0.0	33.3	66.7	0.0	100.0	100.0
interview-dialog.0003_speaker_9	5	63	65.1	31.7	3.2	7.9	42.9	100.0
Sum/Avg	217	1236	63.8	23.3	12.9	5.3	41.4	54.4
Mean	21.7	123.6	39.7+	28.3+	32.0+	4.8+	65.1+	82.9
S.D.	30.9	212.5	31.4+	12.9+	34.1+	4.0+	29.2+	28.3
Median	7.0	57.5	40.7+	32.5+	15.4+	3.2+	62.0+	100.0

5.2.2 Evaluation data

For the evaluation of the transcription results, reference files are needed. They are produced manually, by listening to the audio and writing everything which has been said. It also holds for incomplete words, repetitions and self-corrections the speakers do. We use the software Transcriber[19] for this task. It allows annotating:

- Speech sequences with detailed transcription, speaker name or ID, speaker gender
- Spontaneous speech events such as hesitation, laughter, cough

[18] http://www.nist.gov/speech/tools

[19] http://sourceforge.net/projects/trans/

- Lexical tags such as words from foreign languages, unknown words (cannot be recognised by the human transcriber), or broken words
- Non-speech sequences such as music, noise and silences
- Event tags such as meeting, interview, dictate etc.

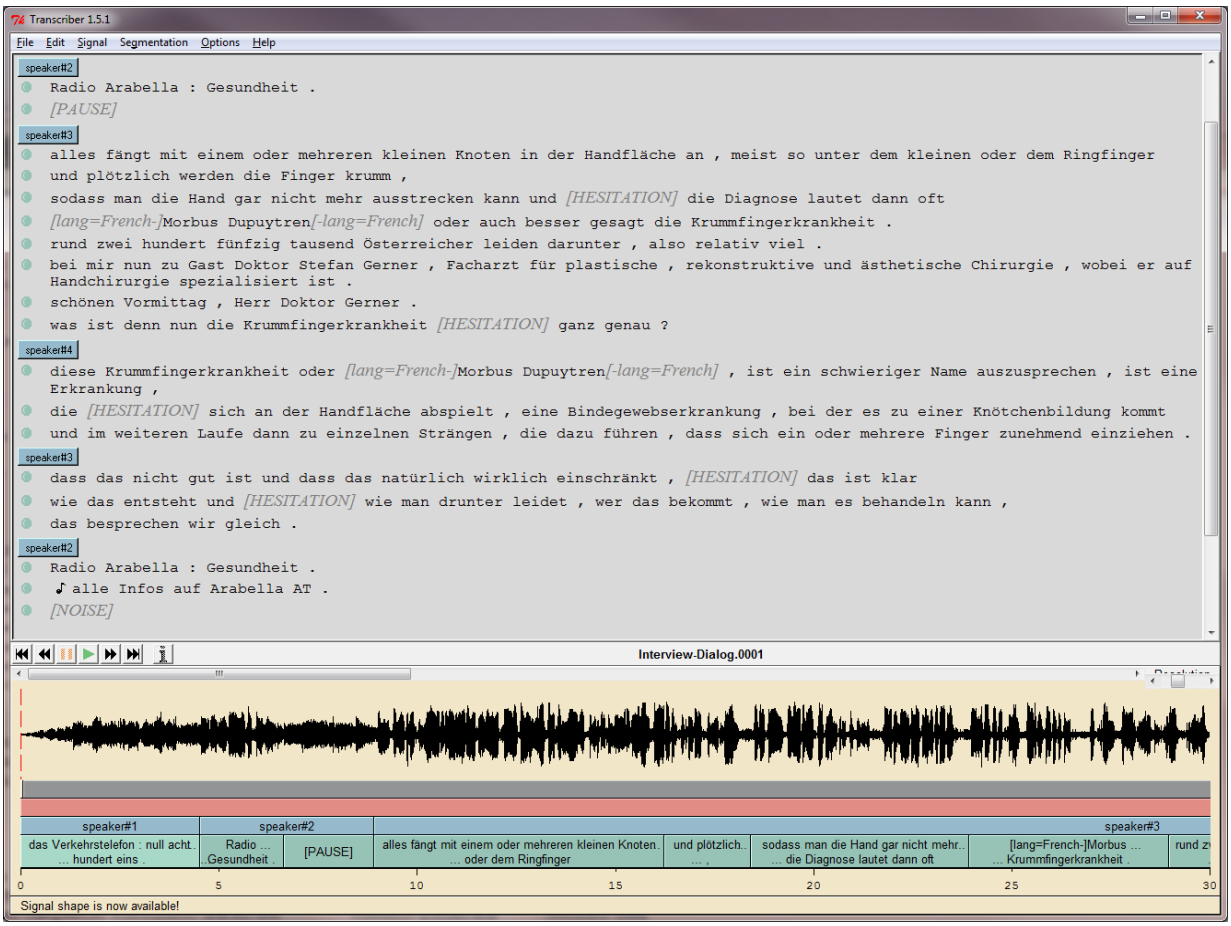

Figure 9: Manual audio transcription in “Transcriber”

We decided to write the reference files in correct orthography (capitalisation) and to put correct punctuation marks as far as possible. Numbers and abbreviations are spelled out. For comparison, the reference files have to be adapted (per scripts), in order for them to correspond to the current development stage and to better match the actual recognition output.

The selected audio corpus for German was collected from different sources. The selection criteria were to equally cover recordings in noisy and in quiet environment, as well as recordings with one or more than one speaker.

The test corpus can be divided into different smaller portions according to different testing purposes and applications, as the need arises.

5.2.3 Evaluation results

After the second year of the project we have achieved the following recognition results for German in the news domain:

Without post-processor the overall WER was 19.3% (see Table 7).

hyp/Nachrichten_no_postproc.txt

SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
n_1	3	655	82.9	13.3	3.8	4.4	21.5	100.0
n_2	1	183	79.8	18.6	1.6	19.1	39.3	100.0
n_3	1	491	88.4	9.0	2.6	2.6	14.3	100.0
n_4	1	261	96.9	3.1	0.0	14.2	17.2	100.0
n_5	1	91	93.4	6.6	0.0	8.8	15.4	100.0
n_6	1	118	90.7	8.5	0.8	11.0	20.3	100.0
n_7	1	404	88.4	10.6	1.0	6.4	18.1	100.0
n_8	1	185	92.4	7.0	0.5	3.2	10.8	100.0
n_9	1	515	83.1	14.0	2.9	2.9	19.8	100.0
Sum/Avg	11	2903	86.9	10.9	2.1	6.3	19.3	100.0
Mean	1.2	322.6	88.4	10.1	1.5	8.1	19.6	100.0
S.D.	0.7	200.1	5.6	4.6	1.4	5.7	8.1	0.0
Median	1.0	261.0	88.4	9.0	1.0	6.4	18.1	100.0

Table 7: Evaluations results for ASR without post-processing

With post-processing we could reduce the overall WER to 15.8%, i.e. an absolute improvement of 3.5% (see Table 8):

hyp/Nachrichten_with_postproc.txt

SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
n_1	3	655	84.6	11.6	3.8	1.7	17.1	100.0
n_2	1	183	80.9	17.5	1.6	16.4	35.5	100.0
n_3	1	491	89.2	8.1	2.6	1.4	12.2	100.0
n_4	1	261	97.3	2.7	0.0	14.2	16.9	100.0
n_5	1	91	98.9	1.1	0.0	0.0	1.1	100.0
n_6	1	118	91.5	7.6	0.8	8.5	16.9	100.0
n_7	1	404	89.9	7.9	2.2	1.7	11.9	100.0
n_8	1	185	93.5	5.9	0.5	2.2	8.6	100.0
n_9	1	515	84.3	12.8	2.9	2.1	17.9	100.0
Sum/Avg	11	2903	88.3	9.4	2.3	4.0	15.8	100.0
Mean	1.2	322.6	90.0	8.4	1.6	5.4	15.3	100.0
S.D.	0.7	200.1	6.1	5.1	1.4	6.1	9.3	0.0
Median	1.0	261.0	89.9	7.9	1.6	2.1	16.9	100.0

Table 8: Evaluation results for ASR with post processing

For English we have not created test data yet. It is planned to use an appropriate test corpus in cooperation with the EU project EUMSSI.

6 MULTIMEDIA CONCEPT DETECTION

This section presents the progress on the techniques applied in multimedia concept detection, which involves the detection of a set of predefined concepts in multimedia files, including videos and images. The steps that comprise the multimedia concept detection procedure are the following:

- Video decoding: this step is applied only in case that the input file is a video and it is responsible for extracting specific frames from the video.
- Feature extraction: this step refers to the extraction of descriptors that describe visually the images/ frames by capturing global or local information.
- Classification: this step is related to the development of models used for classifying images or video frames to the set of predefined concepts/ categories.

In the current deliverable, no progress has been made regarding video decoding. Therefore, in the next section we focus on reporting the work progress for the feature extraction and classification procedures. Moreover, since an overview of the state-of-the-art methods used in the aforementioned three domains (i.e. video decoding, feature extraction, and classification) is provided in the previous deliverable (see section 7.1 – D2.2), we will not go into any further details here.

6.1 Work progress in multimedia concept detection task

6.1.1 Feature extraction

The feature extraction procedure employs methods that aim at describing the visual content of images effectively. The descriptors that are used for representing various image features can be divided into two groups, the global and the local descriptors. These two groups differ in the locality of the feature that is represented. The global descriptors utilize global features of an image, while on the other hand local descriptors represent local salient points or regions. Additionally, in the case of local descriptors, a clustering algorithm is applied to form a vocabulary of "visual words". In the visual word assignment step, the local descriptors are transformed into a "Bag-of-Words" (BoW) representation (Qiu, 2002). Similar keywords are grouped in clusters and each cluster is treated as a visual word that forms the visual vocabulary. Then, the local descriptors are assigned to this vocabulary in a manner that each descriptor is mapped to a visual word. As a result of the aforementioned procedure, a global descriptor that gives an overall impression of visual data is produced.

In the first version of the multimedia concept detection module (see D2.2), strictly local features were extracted. Specifically, the broadly used SIFT (Lowe, 2004) and SURF (Bay et al., 2008) descriptors and their variations, namely RGB-SIFT (Van De Sande et al., 2010), opponent-SIFT,RGB-SURF and opponent-SURF, were used. The visual words assignment was realized using VLAD (Vector of Locally Aggregated Descriptors) encoding (Jegou et al., 2010), a compact and fast to compute approach (Jegou et al., 2012). Moreover, the Principal Component Analysis (PCA) algorithm was applied for dimensionality reduction purposes, following the approach described in (Markatopoulou et al., 2015). The feature extraction procedure can be described as follows: After the extraction of the local descriptors, they were compacted to 80 dimensions for SIFT, SURF, and their variations using PCA and were

aggregated using the VLAD encoding. This resulted in a VLAD vector of 163840 elements for SIFT or SURF. Eventually, the VLAD vectors were compressed into 4000-element vectors by applying a modification of the random projection matrix. These reduced VLAD vectors served as input to the classification step.

In the current version of the multimedia concept detection module, we basically relied on the SIFT descriptor. Firstly, we detected the most significant keypoints in the images and then we extracted visual representations from them. We used the Lip-vireo software[20], which includes a variety of keypoint detection and descriptor extraction algorithms. We chose to use the Fast Hessian keypoint detector (Bay et al., 2008) and extract SIFT features from our datasets. From the already extracted SIFT features, we also computed the corresponding RootSIFT (Arandjelović and Zisserman, 2012) features. The RootSIFT descriptor is an element wise square root of the L1 normalized SIFT vectors and yields superior performance without increasing processing or storage requirements. Both SIFT and RootSIFT descriptors have a dimensionality of 128, so for each image, vectors with *128 x <Number of keypoints>* dimensions were created.

Apart from using only the existing image datasets to train our concept detection models, we also tried to pre-process them before extracting the features and training the models. For each descriptor, we created an extra dataset, which contained the images after performing saliency detection. It is a procedure that aims at finding and isolating the most important information of the image. Specifically, we used the software developed by the Chinese University of Hong Kong[21] to perform hierarchical saliency detection (Yan et al., 2013). Hierarchical saliency detection employs a hierarchical framework that infers importance values from three image layers in different scales, in order to obtain a uniformly high-response saliency map. However, the following should be noted: Multimedia concept detection is applicable only for two MULTISENSOR use cases, namely “Journalism use case scenario” and “Commercial media monitoring use case scenario”. The overwhelming majority of the concepts selected for the commercial media monitoring use case are logo related (i.e. depict logos of well-known home appliances companies). Unlike the experiments conducted in D2.2, we mainly included in the datasets images that contain logos as a small part of them. On the other hand, most of the images regarding the journalistic use case are non-logo related and they depict larger concepts. Due to the fact that it was difficult for the saliency features to successfully capture the logos inside the images (based on initial experiments), we decided to apply the hierarchical saliency detection procedure only to the journalistic use case concepts.

In order to transform the visual descriptors into a single visual representation for each image, we created a vocabulary by using the “Bag-of-Visual-Words” (BoVW) approach. Specifically, we used the traditional BoVW implementation of the Sotu software[22], which utilizes the RB K-Means algorithm to cluster a range of sampled keypoints and form the vocabularies. For each descriptor modality (SIFT, RootSIFT plus saliency-SIFT and saliency-

[20] https://code.google.com/p/lip-vireo/

[21] http://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/

[22] https://code.google.com/p/sotu/

RootSIFT for the journalistic use case concepts) and for each dataset, a vocabulary of 20000 visual words was created. The resulting vector representation of each image had the same dimensionality as the size of the vocabulary. These representations were provided as input to the concept detection classification models.

6.1.2 Classification

In the classification step, concept detection models are developed by using the low-level visual features. Subsequently, image labelling is performed. A suitable classification algorithm is employed, in order to train models separately for each concept on ground-truth annotated corpora. For a new unlabelled image or video shot, the trained models provide confidence scores indicating the belief of each model that the corresponding concept appears in the image or video shot. For each different descriptor used, a new classification model is trained and the prediction results from the models are fused.

In the previous deliverable (D2.2), the algorithm used for classification (i.e. training and testing) was the Support Vector Machines (SVM). SVM performs classification by constructing hyperplanes in a multidimensional space that separates cases of different class labels. For the six descriptors that were extracted, an equal number of models were developed for each concept. The prediction results for each descriptor per concept were fused in a late fusion manner by averaging all classifiers output scores.

In the current deliverable, we employed the same classification algorithm as in the case of D2.2, more specifically the linear SVM version, which can be considered ideal for classifying high-dimensional data, as its maximum margin property gives some guarantees on the generalization performance. The library used for implementing the classification models is LIBSVM (Chang and Lin, 2011). For each concept and for each of the descriptors extracted, a separate classification model was created. Finally, a different late fusion strategy than the one described in D2.2 was used. Specifically, we computed weights for each model's outputs. For each descriptor's classification model, its corresponding training accuracy value was calculated. Then, the accuracy values were normalized (by dividing them by their sum) and served as weights for the models' probability outputs. During the testing phase, for the prediction of an unlabelled image, the probability outputs from the descriptors' models were multiplied by the aforementioned weights and summed to provide the final predictions for each concept.

6.1.3 Concept selection for MULTISENSOR use cases

As already mentioned (see section 6.1.1), multimedia concept detection is applicable only for two MULTISENSOR use cases, namely "Journalism use case scenario" and "Commercial media monitoring use case scenario". For each use case, the concepts that were utilized in the experiments were selected based on two different ways: 1) Visual inspection of exemplary videos / images provided by the user partners and 2) Explicit definition by user partners (e.g. logos of specific companies). The complete lists that were defined for both use cases contain 98 concepts (56 for the journalistic and 42 for the commercial media monitoring use case) and can be found in section 7.4 of D2.2. In the current deliverable, we decided to focus mainly on concepts that were proposed explicitly by the user partners. The 26 concepts (10 for the journalistic and 16 for the commercial media monitoring use case)

that were selected for the experiments regarding the current deliverable are contained in Table 9.

Journalism use case scenario		Commercial media monitoring use case scenario	
• EnBW logo • E-On logo • Nuclear energy logo • RWE logo • Vattenfall logo	• Lattice tower • Recycle bin • Smog • Solar panel • Wind turbine	• Bauknecht logo • Bosch logo • Electrolux logo • General Electric logo • Hoover logo • Indesit logo • LG logo • Miele logo	• Philips logo • Samsung logo • Siemens logo • Whirlpool logo • Clothes Washing Machine • Coffee Maker • Refrigerator • Vacuum Cleaner

Table 9: Selected concepts for the "Journalism" and "Commercial media monitoring" use cases.

6.2 Evaluation

In this section, we present the evaluation of the models developed for the set of concepts described in section 6.1.3. It should be noted that since we don't report any progress regarding video decoding in this deliverable, we will not provide evaluation results on any performance aspects of that specific procedure.

6.2.1 Creation of training datasets

The overall procedure for creating the training datasets that were used in the development of the models for the selected concepts is described in detail in deliverable D2.2 (see section 7.5 and Appendix C1). In a nutshell, the procedure involved the gathering of images related to each specific concept through the use of the Bing Image API[23], the flickr API[24] and Google Images[25], followed by the manual annotation of the images returned by the APIs. However, there is key difference between the procedure followed in D2.2 and the one that was followed in the current deliverable: For each concept in D2.2, the corresponding model was trained using the images related to that concept as positive (relevant) cases, while the images related to the rest of the concepts served as negative (irrelevant) cases. On the other hand, in this deliverable we enriched the group of negative cases for each training dataset with "noise", meaning that we inserted images either referring to logos / concepts not among the selected ones or having totally irrelevant content. Moreover, as already mentioned (see section 6.1.1), we made sure that the training datasets for the logo related concepts contained not only images that depict plain logos (like in the case of D2.2), but also images that include logos as a small part of them. From all the aforementioned, it easily follows that the task of efficiently training the classification models for the selected concepts is more difficult and demanding, compared to D2.2.

[23] https://datamarket.azure.com/dataset/bing/search

[24] https://www.flickr.com/services/api/

[25] https://images.google.com/

6.2.2 Evaluation results

For each of the two MULTISENSOR use cases that multimedia concept detection is applicable for, a test dataset was created, either by retrieving images from the internet or by collecting images provided by the user partners. These test datasets were manually annotated and three popular IR metrics (precision, recall and F-score) were utilized, in order to evaluate the classification performance for each concept separately.

Commercial media monitoring use case scenario

In the commercial media monitoring use case, the test dataset consists of 400 images in total (20 relevant images for each of the 16 concepts plus 80 images containing "noise", in the sense described in section 6.2.1). The classification performance results are depicted in Table 10. It can be said that although the classification models are tested against a demanding dataset, which includes images containing logos as part of the image and images that are characterized as "noise", they achieve quite satisfactory results, especially for the F-score metric, which considers both precision and recall. The macro-average F-score value for the whole set of concepts is approximately 0.67. On the other hand, classification experiments conducted on the same test dataset using the previous multimedia concept detection framework described in D2.2 achieved a macro-average F-score value of around 0.30. This surely indicates a substantial improvement in classification performance and demonstrates the superiority of the feature extraction approach proposed in the current deliverable. Additional experiments on similar datasets using advanced techniques related to object detection will be performed and reported in the next deliverable (D2.4).

Concept name	Number of relevant images	Number of retrieved images	Number of retrieved relevant images	Precision	Recall	F-score
Bauknecht logo	20	11	11	1	0.55	0.709
Bosch logo	20	9	8	0.888	0.4	0.551
Clothes Washing Machine	20	30	17	0.566	0.85	0.68
Coffee Maker	20	60	19	0.316	0.95	0.475
Electrolux logo	20	25	14	0.56	0.7	0.622
General Electric logo	20	18	16	0.888	0.8	0.842
Hoover logo	20	13	13	1	0.65	0.787
Indesit logo	20	17	12	0.705	0.6	0.648
LG logo	20	11	9	0.818	0.45	0.58
Miele logo	20	17	11	0.647	0.55	0.594
Philips logo	20	18	14	0.777	0.7	0.736
Refrigerator	20	33	17	0.515	0.85	0.641
Samsung logo	20	19	16	0.842	0.8	0.82
Siemens logo	20	10	8	0.8	0.4	0.533
Vacuum Cleaner	20	30	19	0.633	0.95	0.76
Whirlpool logo	20	17	13	0.764	0.65	0.702
Macro-average	**-**	**-**	**-**	**0.732**	**0.678**	**0.667**

Table 10: Evaluation metrics values for the "Commercial media monitoring use case" dataset.

Journalism use case scenario

The same approach as in the previous use case was followed for the journalistic use case, in order to create the test dataset. We collected 20 relevant images for each of the 10 concepts and we supplemented the test dataset with 100 "noisy" images, for a total of 300 test images. The results of the evaluation can be seen in Table 11. We observe that for the majority of the concepts, the trained classification models are performing quite well. Moreover, it can be stated that the non-logo related concepts (especially the concept "wind turbine") have benefited from the application of the hierarchical saliency detection, in addition to the extraction of the SIFT and RootSIFT local feature descriptors. Using the framework described in this deliverable, we achieve a macro-average F-score value of 0.75 for the selected concepts, which considerably outperforms the corresponding macro-average F-score value achieved by the previous framework (D2.2) on the same test dataset (around 0.35). Similarly to the commercial media monitoring use case, we will conduct experiments with advanced techniques related to object and event detection and we will present the results in deliverable D2.4.

Concept name	**Number of relevant images**	**Number of retrieved images**	**Number of retrieved relevant images**	**Precision**	**Recall**	**F-score**
EnBW logo	20	12	10	0.833	0.5	0.625
E-On logo	20	12	11	0.916	0.55	0.687
Nuclear energy logo	20	15	15	1	0.75	0.857
RWE logo	20	12	11	0.916	0.55	0.687
Vattenfall logo	20	11	11	1	0.55	0.709
Lattice tower	20	14	13	0.928	0.65	0.764
Recycle bin	20	15	11	0.733	0.55	0.628
Smog	20	34	19	0.558	0.95	0.703
Solar panel	20	18	16	0.888	0.8	0.842
Wind turbine	20	20	20	1	1	1
Macro-average	-	-	-	**0.877**	**0.685**	**0.750**

Table 11: Evaluation metrics values for the "Journalism use case" dataset

7 MACHINE TRANSLATION

In D2.2 we described the technical framework and our approach to the machine translation in MULTISENSOR, and the basic techniques to develop baseline MT systems. We also gave information on data sources collected and used for the training of those systems, and described the crucial steps and tools in the process of data preparation. Regarding the evaluation, we outlined our planned approach. In the first year of the project we had released baseline systems for all languages (German, French, Bulgarian, and Spanish) into English.

Since then we have released the MT systems for English into the other languages and have started working on more advanced versions. In the current document we describe our strategies for improving the translation quality, and thus present the first steps towards the advanced MT systems for MULTISENSOR. In this report we also address the evaluation process in more detail. We list the metrics selected for the MT quality evaluation and describe the development of the evaluation tools for this task. We also give an overview of evaluation results so far.

7.1 Work progress in machine translation task

The manual inspection and error analysis of baseline translations have shown that the error sources occur on different levels, some caused by noisiness in the training corpus, some much more complex and demanding more advanced strategies to be solved. During the reporting period, we concentrated on the reduction of noise in the corpus, retraining, and also modifying the translation parameters accordingly. We have achieved measureable improved results in translation regarding to the following issues:

- Better homogenisation of the training corpus
- Reduction of the amount of unknown words
- Tuning of the model parameters for better translation quality

7.1.1 Data homogenisation

The error analysis had revealed a couple of errors in baseline systems which could be fixed by homogenisation and normalisation of data originating from different sources, as well as homogenisation of training/tuning/testing datasets. In particular, the translation quality could be improved through measures such as the following:

- Consistent general truecasing in all corpora: For the baseline systems we had followed the approach to truecase only the sentence beginnings, but not to touch the body of the sentences. However, texts contain capitalised words also within the sentences, which leads to an unnecessarily high number of different tokens (the, The) and their contexts. This has been changed for the current version.
- Reduction of spelling variants to one “standard term”, e.g.: double quotes (“”„“»«»), same words with or without hyphen in English, old/new orthography in German etc.
- Consistent tokenisation of e.g. contractions (it’s)and web-addresses, to name only two.

7.1.2 Reduction of unknown words

Translation quality can be strongly impacted by unknown words. Missing translations for single words or phrases can substantially determine the translation of the surrounding words and thus of the entire sentence context.

In many experiments, we have observed two different types of unknown words in the translation: The first ones are words that have not been seen in the source part of the training corpus at all, and as such they cannot have any corresponding translations in the target language. The second type, which is more difficult to trace, are words that occurred in the training corpus, but the word and phrase alignment components only were able to find translations for them as part of bigger phrases, but not as single words and consequently not in other contexts.

In the first case, we first decided to not train the models again, but to enrich the phrase tables by adding translations for unknown words to them. Since these words were not existent in the phrase tables at all, they could simply be added with a default weight, equal for all of them. This helped to reduce OOV problem substantially. The evaluators' subjective impression was that the translation is much better than before, even if the gainin the translation quality measured automatically was relatively low. Important shortcomings of this approach are, however, the lack of the context information and missing inflected forms for the imported translations.

In the second case, the unknown words have been seen in the training corpus, and as such they are part of some bigger (weighted) phrases, but however, the system was not able to automatically discover and learn single translations for them. Since in this case a manual manipulation of the phrase tables was not an option (there exist weighted phrases containing these words), we decided to retrain the models and to try to influence the alignment process. For this, we added to the training corpus many single lexicon entries and also many selected, very short and simple expressions and sentences from available translation memories. In this way, by adding simple and unambiguous lexical translations to the training corpus, many correct single word alignments could be learned already during the initial alignment iteration, which increased the overall word and phrase alignment quality. Later experiments have shown that this approach can also be applied to the first type of unknown words with very similar results as the approach described above.

7.1.3 Tuning for quality

A machine translation system is composed of several statistical models, such as bilingual translation models, reordering models, and monolingual language models. After the training, all of them are used with equal weights in the translation process. By tuning and optimising their weights the translation can be customised and improved to some extent for specific translation requirements such as different domains and vocabularies. For this, a trained system is being tuned by translating a representative dataset in several iterative testing and adjusting cycles, in order to reassess the parameters for the respective purpose (Bertoldi et al., 2009). The tuning dataset, also called development set, can be a relatively small parallel corpus, but it should be as domain and application-specific as possible. Our experiments have shown that already tuning with very small datasets can be a very effective means for

domain adaptation, in particular if the available in-domain data are too sparse for a domain-adaptive retraining of the whole system. Important is to keep this small parallel corpus separate from the training corpus (“unseen” during the training), to not bias the system.

For the first domain-tuning cycle in MULTISENSOR, no parallel data related to the topics energy (UC1), household appliances (UC2) and dairy products (UC3) were available, thus we decided to experimentally tune the system to the “journalistic domain in general”, by using an open-source news corpus compiled for the WMT competitions[26].Only for Bulgarian, being not a part of the WMT tasks until now, we used the SEtimescorpus (Tiedemann, 2009) for both training and tuning, by separating it in respective portions. We started our experiments with up to 13.000 sentences in German-English and observed that a certain convergence is reached already by using 3.000 sentences, if they are from the same or similar domain. That is why we finally used tuning sets of 3.000 sentences for each language direction.

7.2 Evaluation

As the means to measure MT quality in MULTISENSOR we apply both automatic evaluation and manual evaluation. The assessment of the translation quality after each development step and comparison of intermediary versions is being done automatically. Manual human evaluation, being very labour-intensive and expensive, will be performed in the final phase of the MT development. In the second year of the project we prepared appropriate test data and developed the evaluation tools needed for the manual evaluation.

7.2.1 Evaluation tools

Automatic evaluation

For the automatic assessment of the translation quality we have evaluated the open-source tools BLEU (Papineni et al., 2002), NIST[27], and METEOR (Denkowski and Lavie, 2011) and decided to use BLEU, which is a widely used standard in the MT community. It measures how similar is the MT output to one or several reference translations ("the closer a machine translation is to a human translation, the better it is").We use this automatic metric as an efficient technique to monitor progress in the development of different MT versions.

Manual evaluation

However, the automatic metrics can neither reliably measure the actual translation quality, nor can they assess the user satisfaction. For these reasons, additional manual evaluation is needed, in order to form an actual user opinion on the translation quality. We follow these two usual approaches in manual evaluation:

- Users compare different translations of a sentence and decide which one is better/worse (“comparative evaluation”)
- Users compare only one translation with its source sentence and give their opinion on adequacy and fluency of the translation (“absolute evaluation”)

[26] http://www.statmt.org/

[27] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a-20091001.tar.gz

For the manual evaluation task, Linguatec has developed a test frame with two different GUIs for comparative/absolute evaluation (see Figure 10).The main functionality of the tools is:

- Import of a new evaluation 'package' (sentences and translations to be compared)
- Interactive support of the evaluation procedure
- Creation of result files containing statistics

The main files are the imported source and target sentences, as well as the evaluation xml files, where the results of the evaluation are stored. An overview file can be created containing basic statistics.

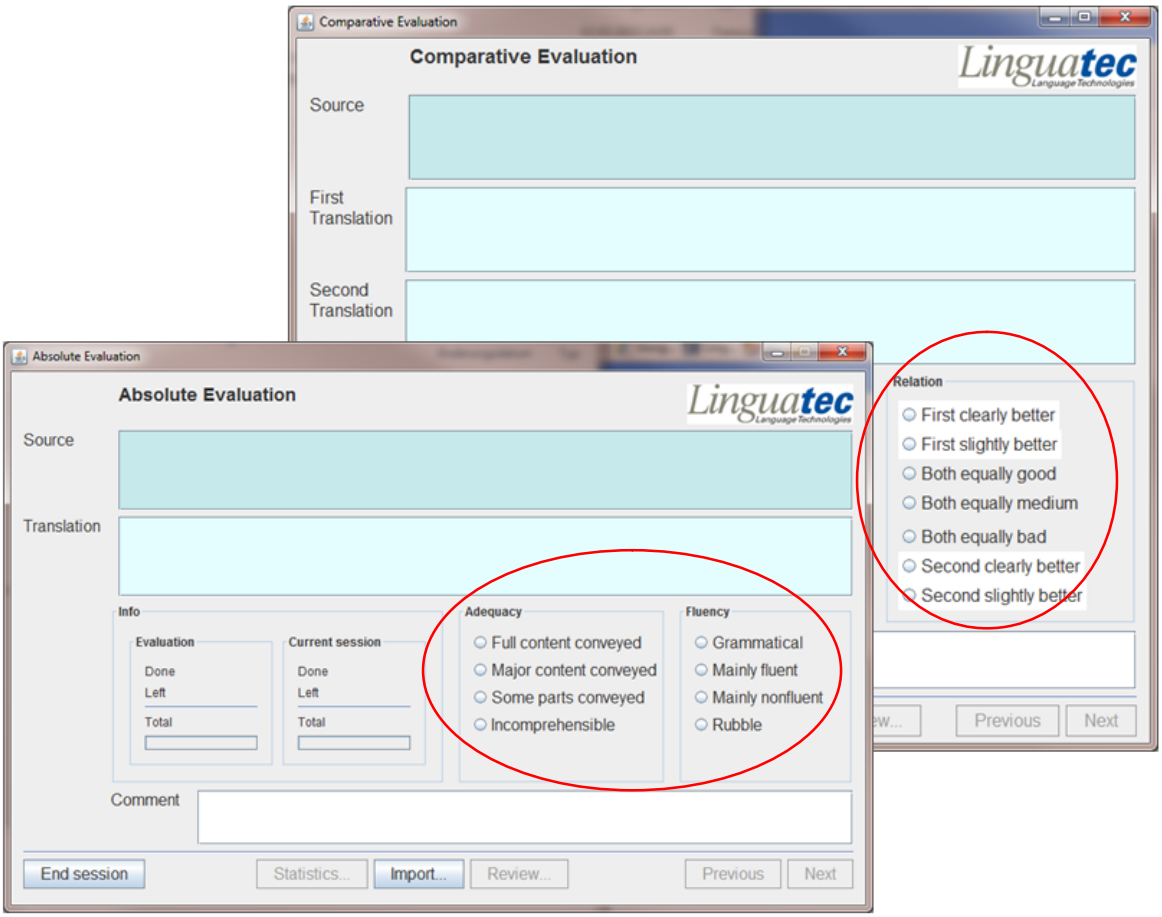

Figure 10: Tools for manual MT evaluation

7.2.2 Evaluation data

The test sentences must be seen neither in the training corpus nor in the development (tuning) data set. Best results can be achieved in translating test sentences of a similar domain to the training data. For the comparison, the reference file and system output have to be sentence-aligned (each line in the reference file corresponds to a line in the system output). Both texts need to be tokenized (punctuation marks detached) and truecased in the same way as the training corpora.

For MULTISENSOR, we have been using a test sets from the journalistic domain, taken from the same sources like the development sets (SEtimes for Bulgarian, and WMT competition data for all other languages). They contain 3.000 sentences for each language directions.

7.2.3 Evaluation results

Table 12 presents the results of the MT development cycles done so far, in particular for the baseline systems and the first three adaptation steps, as described above. The results are presented in BLEUscores, measured by using the multi-bleu.pl[28] script. All results refer to the test sets described above. The multi-bleu.pl script allows comparison of the output and the reference files as they are, i.e. truecased (TC) or in all-lowercase format(LC) for better comparison. For Bulgarian, it does not offer this functionality. The lowercased comparison shows generally better results, since it ignores differences in capitalisation only.

	S0 (baseline)		S1 (homogenised orthography)		S2 (reduced unknown words)		S3 (tuned for the "general journalistic domain")	
	TC	LC	TC	LC	TC	LC	TC	LC
bg-en	22.48	24.04	24.87	25.99	25.99	26.01	27.11	28.99
de-en	18.09	18.81	18.81	18.90	18.90	18.91	18.80	19.57
es-en	19.73	20.33	20.66	20.93	21.00	21.02	23.92	24.95
fr-en	19.64	20.29	20.28	21.34	21.54	21.99	23.31	24.13
en-bg	22.18	-	22.16	-	22.16	-	22.18	-
en-de	14.50	14.71	14.71	14.81	14.81	14.81	14.90	15.13
en-es	23.19	23.97	23.90	23.91	23.93	23.94	23.19	23.97
en-fr	23.54	24.08	24.05	24.01	24.05	24.08	23.54	24.08

Table 12: MT evaluation results measured in BLEU

7.2.4 Web-based demo frame

The evaluation tools presented above are supposed to serve as a testing environment for the final quality assessment by selected human evaluators, which will take place towards the end of the project.

But, already during the development of the MT engines, and especially for the 1st project review, a requirement was defined to make the MT functionality available for an easy test access and for demo purposes. For this, a web-based test platform has been created. It can be accessed through the following URL: http://services.linguatec.org/html/translate.

[28] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

The very simple GUI contains a pull-down list with the available language directions, two editable text fields for the source text and for the translation, and a “Translate!” button. There is no limitation to the number of characters/words to be translated. In the Google Chrome browse the size of the text fields is scalable and can be adjusted to the size of the text, while some other browsers do not support this function.

8 LANGUAGE IDENTIFICATION MODULE

8.1 Approach

This module determines in which natural language a text is written. There are different approaches to language identification, statistical and rule (knowledge) based ones.

The module developed for MULTISENSOR is knowledge based, i.e. it uses rules and lexical resources to guess the language of the given text or string. It works robustly even on short text pieces (15-20 characters) and closely related languages, like Dari and Farsi, which differ only in some lexical entries.

The programme code is written in Java. Initially, text preprocessing steps are done, such as tokenisation and normalisation. The corelanguage identification is done in three main steps: (a) classification according to the major Unicode character blocks: Cyrillic, Greek, Arabic etc. (b) refinement within a block and the definition of exceptions on character level (c) consultation of lexicons and lists of most frequent words in each language for final decision. Scores for hypotheses are collected and ranked. For the project, there is a decision to store only the language with the highest score. If necessary, this can be changed to provision of many hypotheses with confidence weights.

8.2 Languages

It has been designed to recognise as precise as possible the project languages: Bulgarian (bg), English (en), French (fr), German (de), and Spanish (es). It also recognises, for better discrimination purposes, 18 other languages: Arabic (ar), Chinese (zh), Czech (cz), Danish (da), Dutch (nl), Finnish (fi), Greek (el), Hebrew (he), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Pashto (ps), Polish (pl), Portuguese (pt), Russian (ru), Swedish (sv), and Turkish (tr). All other languages get the tag “unknown”.

8.3 Integration into the MULTISENSOR platform

The main purpose in the project is to annotate incoming texts with a language tag before storing them into the central repository for further analysis.

The language identification module runs on Linguatec servers. The communication between the module and MULTISENSOR platform is established via REST web services. The module gets only the text as a string as the only parameter, and returns a 2-characters language code according to ISO 639-1. There are no dependencies with other modules.

For testing purposes, the REST service endpoint can be used:
http://services.linguatec.org/rest/lang/identify?text=[text] ([text] is encoded).

9 CONCLUSIONS

This deliverable reports on advanced techniques and the continuation of the development of the modules for named entities recognition, concept extraction from text, concept linking and relations, speech recognition, multimedia concept detection, and machine translation.

The **named entities recognition** module has been extended from a basically lexicon-based to a contextual coverage, and adapted to the use cases of the MULTISENSOR project. Both actions have brought a significant better recall and precision for the recogniser. For tests, about 60 documents of the use cases have been collected and marked up for further comparative evaluation. Similar activities for a broader lanuage coverage have been launched. The next deliverable will focus on the extension of languages, and a further increase of recognition quality for the existing use cases.

The **concept extraction** and **concept linking and relation extraction** modules have also been improved. A multilingual deep-syntactic parser that outputs syntax-oriented predicate-argument structures (English and Spanish so far) has been developed. The Stanford Coref tool has been adapted to recognize the annotations of entities and concepts produced by the NER and concept extraction service as potential coreferring mentions. A new rule-based frame-semantics parser has been set up for English, which allows for predicate-argument identification while ensuring the suitability of the output for the natural language generation pipeline (this new tool is portable to other languages with the appropriate resources). A gold standard annotation of frame-based structures has been produced for evaluation purposes. Finally, a model used to integrate FrameNet RDF and NIF has been developed. In the final deliverable (D2.4), the final version of these modules will be described and evaluated, and their performance according to the indicators will be assessed.

As far as the **speech recognition** module is concerned, German has been released and many post-processing steps towards the advanced version for both languages German and English have been done. In particular, we have worked on the reconstruction of orthography (correct casing instead of all-lowercase), reconstruction of numbers and symbols (their most usual writing form instead of the spelled one), insertion of utterance boundaries (breaking up the long recognition string into shorter phrases for better text analysis), and on joining compound parts into whole words. Furthermore, 14 hours of audio data in German has been manually transcribed in order to create a representative test set. First evaluation cycles have been done. In-domain adaptation for the three project use-cases will be done as soon as the data in the news repository are available, tagged correctly by the use-case and language.

Regarding the **multimedia concept extraction** module, in this deliverable we presented a framework, where alternative local descriptors (RootSIFT, saliency-SIFT and saliency-RootSIFT) are utilized in its feature extraction step (compared to the corresponding framework in D2.2). Furthermore, a different late fusion strategy is applied in the classification step. Finally, quite realistic datasets are used, both for training and testing purposes (with images that can be identified as “noise” and complicated images for the logo related concepts). In general, the classification models achieved an adequate performance for the majority of the employed concepts. In the next deliverable (D2.4), advanced object detection techniques that have the ability to locate specific objects within an image will be presented and tested on similar realistic datasets.

As for the **machine translation** module, all language directions have been released. Many tuning and post-processing steps have been done towards the advanced versions of SMT. In particular, we have worked on the data homogenisation (better normalisation and tokenisation), reduction of unknown words, and tuning for better translation in the general journalistic domain. Furthermore, we have compiled the test sets, and have performed first cycles of the automatic MT evaluation. For the manual evaluation, which will be done in the end of the project, tools with user interfaces have been developed. Additionally to these evaluation tools, a web-based test frame has been provided.

The **Language identification** module has been developed in order to meet the need for annotating all incoming texts by correct language code, before storing them into the news repository. The module supports the identification of all project languages, and also of several other languages, for better discrimination. Not recognised languages are marked by "unknown". The component runs on Linguatec servers and communicates with the MULTISENSOR server via REST API.

REFERENCES

Albatal, R., and Mulhem, P. (2010): “MRIM-LIG at ImageCLEF 2010 Visual Concept Detection and Annotation task”, In CLEF.

Arandjelović, R., and Zisserman, A. (2012): “Three things everyone should know to improve object retrieval”, 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2911–2918.

Baber, J., Afzulpurkar, N., and Bakhtyar, M. (2011): “Video segmentation into scenes using entropy and surf”, 7th International Conference on Emerging Technologies (ICET), pp. 1–6.

Ballesteros, M., Bohnet, B., Mille, S., and Wanner, L. (2015): “Data-driven Deep-Syntactic Dependency Parsing”, Natural Language Engineering, Cambridge University Press , p. 1-36.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008): “Speeded-up robust features (surf)”, Comput. Vis. Image Underst., vol. 110(3), pp. 346–359.

Bertoldi, N., and Mederico, M. (2009): “Domain adaptation for statistical machine translation with monolingual resources”, Fourth workshop on statistical machine translation, pp. 182 – 189.

Bohnet, B., and Wanner, L. (2010): “Open source graph transducer interpreter and grammar development environment”, Proceedings of the 7th International Conference on Language Resources and Evaluation, p 211-218.

Bohnet, B., and Kuhn, J. (2012): “The Best of Both Worlds–A Graph-Based Completion Model for Transition-Based Parsers”, Proceedings of the Biannual Meeting of the European Chapter of the Association for Computational Linguistics, p. 77-87.

Bohnet, B., and Nivre, J. (2012): “A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing”, Proceedings of the Conference on Empirical Methods in Natural Language Processing, p. 1455-1465.

Bosma, W., and Vossen, P. (2010): “Bootstrapping Language Neutral Term Extraction”, Proceedings of LREC 2010.

Chang, S.-F., Sikora, T., and Purl, A. (2001): “Overview of the MPEG-7 standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 11(6), pp. 688 –695.

Chang, C. C., and Lin, C. J. (2011): “LIBSVM: A library for support vector machines”, ACM Transactions on Intelligent Systems and Technology (TIST), 2(3), 27.

Charniak, E. (1997): “Statistical Parsing with a Context-free Grammar and Word Statistics”, Proc AAAI.

Das, D., Schneider, N., Chen, D., and Smith, N. A. (2010): “SEMAFOR 1.0: A probabilistic frame-semantic parser”, *Language Technologies Institute, School of Computer Science, Carnegie Mellon University*.

Denkowski, M., and Lavie, A. (2011): “Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems”, Proceedings of the Sixth Workshop on Statistical Machine Translation, pages 85–91.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015): “Transition-Based Dependency Parsing with Stack Long Short-Term Memory”, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference of the Asian Federation of Natural Language Processing, p. 334-343.

Fillmore, C. J., Baker, C. F., and Sato, H. (2002): “The FrameNet Database and Software Tools”, In LREC.

Jegou, H., Douze, M., Schmid, C., and Perez, P. (2010): “Aggregating local descriptors into a compact image representation”, In: IEEE on Computer Vision and Pattern Recognition (CVRP 2010). pp. 3304-3311. San Francisco, CA.

Jegou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P., and Schmid, C. (2012): “Aggregating local image descriptors into compact codes”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34(9), pp. 1704-1716.

Johansson, R., and Nugues, P. (2007): “Extended constituent-to-dependency conversion for English”, Proceedings of 16th Nordic Conference of Computational Linguistics, p. 105-112.

Kingsbury, P., and Palmer, M. (2002): “From Treebank to PropBank”, Proceedings of the Third International Conference on Language Resources and Evaluation, p. 1989-1993.

Lowe, D. G. (2004): “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, vol. 60, pp. 91–110.

Markatopoulou, F., Pittaras, N., Papadopoulou, O., Mezaris, V., and Patras, I. (2015): “A Study on the Use of a Binary Local Descriptor and Color Extensions of Local Descriptors for Video Concept Detection", Proc. 21th Int. Conf. on MultiMediaModeling (MMM'15), Sidney, Australia.

McDonald, R., Pereira, F., Ribarov, K., and Haji˘c, J. (2005): “Non-Projective Dependency Parsing Using Spanning Tree Algorithms”, Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, p. 523–530.

Mel’čuk, I. (1988): “Dependency Syntax: Theory and Practice”, State University of New York Press.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., and Grishman, R. (2004): “The NomBank Project: An Interim Report”, Proceedings of the Workshop on Frontiers in Corpus Annotation, p. 24-31.

Mille, S., Burga, A., and Wanner, L. (2013): “AnCora-UPF: A Multi-Level Annotation of Spanish”, Proceedings of the Second International Conference on Dependency Linguistics, p. 217-226.

Nivre, J., Hall, J., K˙ubler, S., McDonald, R., Nilsson, J., Riedel, S., and Yuret, D. (2007): “The CoNLL 2007 Shared Task on Dependency Parsing”, Proceedings of the CoNLL-ST-07, p. 915-932.

Nuzzolese, A. G., Gangemi, A., and Presutti, V. (2011): “Gathering lexical linked data and knowledge patterns from framenet”, Proceedings of the sixth international conference on Knowledge capture (pp. 41-48). ACM.

Palmer, M. (2009): “Semlink: Linking propbank, verbnet and framenet”, Proceedings of the Generative Lexicon Conference, p. 9-15.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002): “BLEU: a method for automatic evaluation of machine translation”, Proceedings of the 40th annual meeting of the association for computational linguistics, pp. 311-318.

Qiu, G. (2002): “Indexing chromatic and achromatic patterns for content-based colour image retrieval”, Pattern Recognition 35, pp. 1675-1686.

Schuler, K. K. (2005): “VerbNet: A broad-coverage, comprehensive verb lexicon”, Ph.D. thesis, University of Pennsylvania.

Siegler, M., Jain, U., Raj, B., and Stern, R. (1997): “Automatic Segmentation, Classification and Clustering of Broadcast News Audio”, Proceedings of the DARPA Speech Recognition Workshop, p. 97-99.

Tiedemann, J. (2009): “News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces”, Recent Advances in Natural Language Processing (vol V), p. 237-248.

Van de Sande, K., Gevers, T., and Snoek, C. (2010a): “Evaluating color descriptors for object and scene recognition”, IEEE Trans. Pattern Anal. Mach. Intell., vol. 32(9), pp. 1582–1596, September 2010a.

Yan, Q., Xu, L., Shi, J., and Jia, J. (2013): “Hierarchical saliency detection”, 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1155-1162.

A Appendix: Annotation guidelines

This document describes a set of guidelines for the annotation of texts with semantic data and associated lexical information. The annotation has the goal of facilitating the development and evaluation of advanced Information Extraction (IE) and Automatic Summarization (AS) techniques.

The annotation procedure is divided into four separate subtasks:

1. Annotation of expressions denoting ***Named Entities*** (NEs)
2. Annotation of expressions denoting **concepts** relevant to the domain.
3. Annotation of **coreference relations** between expressions that refer to the same NE
4. Annotation of relations indicated by linguistic **predicates**

The annotation is done according to a set of resources which act as reference bodies of knowledge: **BabelNet**[29] for entities and concepts, and **VerbNet**[30] and **FrameNet**[31] for predicates. In the case of predicates, we annotate both predicative senses and their arguments together with the roles taken by these arguments. Two separate annotations of predicates and their arguments are foreseen, one adopting FrameNet as the repository of predicative senses, and another one adopting and VerbNet.

A.1. Annotation of named entities

We annotate with the label **Entity** nominal expressions in the text that refer to named entities belonging to any of the following categories:

- **Person**: individual described by first and last name and, in some cases, gender and occupation (e.g. *Manfred Fischedick*).
- **Organisation**: companies, institutions or other private entities (e.g. *Electrolux*, *European Commision, Wuppertal Institute*).
- **Location**: countries, cities, continents, streets, as well mountains, continents, etc. (e.g. *Germany, Europe*).
- **Product/Brand**[32]: thing made or manufactured, commercial nom of this product (e.g. *Electrolux, Miele, Danone*).
- **Time**: dates, hour descriptions and also relational information that positioned and event in a point of time (e.g. *year, 2013, three days ago*).
- **Unit**: one amount and its corresponding unit (e.g. *20%, 5 kilometres*).
- **Communication**: URLs, email addresses, phone and fax numbers.

By nominal expressions we mean nouns and sequences of words with nouns as their head.

[29] http://wiki.dbpedia.org/
[30] http://verbs.colorado.edu/verb-index/index.php
[31] https://framenet.icsi.berkeley.edu/fndrupal/framenet_search

[32] Sometimes there is ambiguity and a NE labeled as PRODUCT/BRAND are difficult to distinguish from a COMPANY NAME.

Named entities are annotated in two stages: annotation of nominal expressions and annotation of their reference. First we annotate nominal expressions with the label **Entity** according to the following rules:

- Annotate all nominal expressions which the annotator can infer to belong to one of the above categories.
- Exclude determiners, possessives and modifiers from the annotation.
- E.g. in *Professor Dr Manfred Fischedick* annotate *Manfred Fischedick*
- If a complex nominal expression is annotated as a single entity, do not annotate any of its parts again as separate entities.
- E.g. in *United States of America* do not annotate *America* as a separate entity

Entity annotations are assigned with the id of a **BabelNet**[33] synset, if a suitable one can be found. The search is conducted using the normalization options in the annotation form. The following criteria are used:

- Search the text annotated and choose from the set of BabelNet synsets returned by Brat.
- Edit the text in the search dialogue to find the right sense, if necessary. For instance:
  - Remove plural marks
  - Replace white spaces with underscores
  - Replace with equivalent expression: “North American Region” to “North_America”

A.2. Annotation of concepts

We annotate with the label **Concept** those nominal expressions that refer to concepts that do not fall in any the categories of NEs. By nominal expressions we mean nouns or sequences of words with a noun as their head, thus excluding adjectives and other grammatical categories.

We consider for annotation all nominal expressions which refer to concepts except those that are used for rhetorical purposes, e.g. figures of speech. Terminologists often classify candidate terms into the following categories:

1. Terms idiosyncratic to the domain at hand, e.g. *renewable energy* in the energy policies domain.
2. Terms that belong to other domains related to the domain at hand.
3. Generic terms which refer to highly abstract concepts that do not belong to any specific domain.

We annotate all expressions which refer to concepts falling in any of the categories above, but differentiate those falling into the first category by adding a special mark.

The annotation described in this section also foresees two stages: annotation of nominal expressions and annotation of their reference. Annotation of nominal expressions proceeds according to the following rules:

[33] http://babelnet.org/

- Annotate nominal expressions falling in any of the three categories outline above.
- Mark those annotations that, according to the criterion of the annotator, denote domain-specific concepts. Mark this by adding the character '*' to the Notes section of the annotation.
- Do not include determiners or possessives introducing nominal expressions in their annotation.

If a suitable **BabelNet**[34] synset can be found for a concept annotation, it is assigned to it. Synsets can be searched and assigned through the normalization options in the annotation form. The following criteria are used:

- Search the text annotated and choose from the set of BabelNet synsets returned by Brat.
- If no suitable synsets are found, edit the text in the search dialogue. For instance:
  - Remove plural marks
  - Replace white spaces with underscores
  - Expand abbreviated forms, e.g. *dr* to *doctor*
  - Replace with an equivalent expression, e.g. *North American Region* with *North America*
- If no synset is found yet for a multiword expression, annotate as concepts all its parts if they satisfy all of the following constraints:
  - The meaning of the multiword expression if compositional. Otherwise, do not annotate any of its parts.
  - Annotate as concepts parts of the nominal expression which are also nouns or nominal expressions (even if they behave as adjectives) for which a synset can be found.
  - Once a part has been annotated as a concept, do not further annotate any of its subparts.

A.3. Annotation of coreference relations

We annotate coreference relations between expressions that can be inferred to denote the same entity. Coreference relations are annotated by adding **coref** arrows between annotations.

The following cases are foreseen:

- Repetitions of the same expression: Add a **coref link** from the annotation of the new occurrence of the expression to the previous one.
- E.g. Multiple occurrences of term *nuclear power*
- Non-anaphoric expressions which can be inferred to have the same referent as previous expressions: Add a **coref link** from the annotation of the new occurrence of the expression to the previous one.
- E.g. *Germany* and *the country*

[34] http://babelnet.org/

- Anaphoric and cataphoric expressions (pronouns and other pro-forms): Add a **Coref** annotation to the anaphoric expression, and a **coref link** to the last co-referent mention.
- E.g. *Country* and *itself*

Also annotate cases of multiple or partial co-reference:

- *John* and *Peter* like to walk alone. *They* often do.
- **Germany** and **France** oppose the new legislation. **Both** countries demand changes.

A.4. Annotation of predicates

Predicates are annotated in two separate annotations, one according to **VerbNet** and another one according to **FrameNet**. These two resources are semantic repositories of predicative senses and also dictionaries of linguistic predicates associated to senses. Each predicative sense can be realized by one or more linguistic predicates and is associated with a set of arguments that take semantic roles.

In FrameNet senses are referred to as *frames* and their lexical realizations as *lexical units*. Each frame specifies a set of frame elements (FEs) which correspond to arguments identified by their role in the predicate. In VerbNet senses correspond to *verb classes* which have as members the verbs that can be used to realize them. Verb classes have a set of syntactic frames that describe configurations arguments and their roles, along other information.

In the following we describe a general annotation procedure of predicates for both VerbNet and FrameNet, distinguishing between the two when necessary. We foresee two steps, first the annotation of linguistic predicates, and then the annotation of its arguments.

We annotate linguistic predicates according to the following criteria:

- Identify a candidate predicate p and create an annotatiom of type **Predicate** for it:
  - For VerbNet consider **verbs** only.
  - For FrameNet consider also **predicative nouns, semantic prepositions, semantic adverbs, adjectives, coordination particles and discourse connectives**.
  - Include in the annotation of the predicate:
    - Particles part of phrasal verbs.
    - E.g. annotate *shut down* instead only *shut*.
  - Do not include in the annotation of the predicate:
    - Auxiliary or modal verbs.
    - E.g**.** annotate *passed* in *have been passed.*
    - Determiners or modifiers.
    - E.g. annotate *hit* in *hit strongly*.
    - Governed prepositions.
    - E.g. annotate *discussions* in *discussions about*.
- Look up p in FrameNet/VerbNet in order to find a suitable sense s:
  - Use the normalization form of the predicate annotation to lookup up the text being annotated.
  - Choose either FrameNet or VerbNet.
  - In the look-up dialog the text to search can be edited:
    - Inflected verbs can be replaced with their infinitive forms.

        - Plurals can be turned into singular.
        - The grammatical category should not be changed, e.g from adviser to advice.
        - The predicate should not be replaced with a synonym or equivalent expression
    - Choose amongst the senses returned by the search the sense closest to the meaning of the predicate in the text and assign it to the predicate p.
    - If no senses are returned that match the meaning of p, leave it as an unlabelled annotation.

After p has been annotated, proceed with the annotation of its **semantic arguments**. For each text fragment f that corresponds to an argument:

- If f or a part of it (which includes its syntactic head) is marked with an annotation, add an argument arc from p to this annotation.
- If f is not annotated, annotate it with an annotation of type **Arg**, and then add an argument arc from it to the annotation of p.
- If p has a sense s associated to it, determine the role of f from the list of roles/FEs associated to s in either FrameNet or VerbNet. Annotate the role in the Notes section of the arc connecting the argument to the predicate.
    - For VerbNet a role name, e.g. *agent*.
    - For FrameNet a FE, either core or non-core, e.g. *producer*.

Special cases:

- Semantic adverbs are always marked as arguments of the predicates they modify.
- Semantic prepositions are always marked as arguments of the predicates they are associated with.
- Adjectives are always marked as predicates that take as argument the noun they modify.
- In predicates with complex nominal constructions as arguments, mark the head noun as the predicate argument.
- Annotate semantic possessive marks*'s* as unlabelled predicates with unlabelled arguments. Possessive marks are semantic when the sentence stops making sense if they're removed.
- Annotate % signs as unlabelled predicates with unlabelled arguments (e.g. the number preceding the % sign and the expression denoting the total over which the percentage applies).
- Annotate coordinations as as an unlabeled predicate and add unlabeled argument arcs from it to each coordinated element. Instead of linking predicates governing the coordination directly to members of the coordination, add a single link from the governing predicate to the unlabeled predicate standing for the coordination.
- Appositions which do not contain a predicate are annotated by marking the orthographic mark as an unlabelled predicate, adding unlabelled arguments from it.
    - Do not do this for appositions where there is a predicate, e.g *my dog, **in** the sofa*.
    - Do sot for appositions where there is no clear predicate, e.g mark the comma in *my dog **,** a bull terrier* as a predicate.
    - Also mark orthographic marks for appositions in which two predicates don't act as arguments of each other, e.g. *this **is** bullshit, **talking** in a bad way*.