MULTISENSOR

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

FP7-610411

D2.4

Speech analysis, concept extraction and event detection modules

Dissemination level:	Public
Contractual date of delivery:	Month 32, 30/06/2016
Actual date of delivery:	Month 33, 08/07/2016
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 T2.6 Multimedia concept and event detection
Type:	Prototype
Approval Status:	Final Draft
Version:	2.0
Number of pages:	40
Filename:	D2.4_SpeechAnalysisConceptExtractionEventDetection_2016-07-08_v2.0.pdf

Abstract

This deliverable presents the final functionality of the techniques for multimodal content

analysis and their encapsulation in the MULTISENSOR architecture.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	17/05/2016	Outline of the skeleton of the Deliverable	Leo Wanner (UPF)
0.2	17/05/2016	Extended outline	Gerard Casamayor (UPF)
0.3	10/06/2016	Preliminary version of T2.4 description	Simon Mille (UPF)
0.4	16/06/2016	Concept extraction	Joan Codina (UPF)
0.5	17/06/2016	Entity Linking	Gerard Casamayor (UPF)
0.6	18/06/2016	Intro + evaluation of relation extraction	Gerard Casamayor (UPF)
0.7	20/06/2016	Relation extraction + general organization	Alicia Burga (UPF)
0.8	28/06/2016	Integration of LT and CERTH sections + Format	Alicia Burga / Simon Mille (UPF)
0.9	29/06/2016	Revision of the full draft	Leo Wanner (UPF)
0.95	03/07/2016	Internal review	Jens Grivolla (UPF)
1.0, 2.0	08/07/2016	Final version	Leo Wanner (UPF)

Author list

Organization	Name	Contact Information
UPF	Leo Wanner	leo.wanner@upf.edu
UPF	Gerard Casamayor	gerard.casamayor@upf.edu
UPF	Simon Mille	simon.mille@upf.edu
UPF	Miguel Ballesteros	miguel.ballesteros@upf.edu
UPF	Joan Codina	joan.codina@upf.edu
UPF	Alicia Burga	alicia.burga@upf.edu
CERTH	Dimitris Liparas	dliparas@iti.gr
CERTH	Anastasia Moumtzidou	moumtzid@iti.gr
CERTH	Stefanos Vrochidis	stefanos@iti.gr
Linguatec	Reinhard Busch	r.busch@linguatec.de
Linguatec	Gregor Thurmair	g.thurmair@linguatec.de
Linguatec	Boris Vaisman	b.vaisman@linguatec.de

Executive Summary

This report presents the final functionality of the techniques for verbal (spoken and written) and non-verbal (image) content analysis in MULTISENSOR and their encapsulation into individual modules within the MULTISENSOR architecture. Special attention is given to the achievements after the submission of the preceding deliverable on this topic, namely D2.3, in the areas of concept extraction, concept linking and relation extraction and multimedia concept and event detection.

In the area of concept extraction, the work concerned, first of all, the realization of a hybrid (statistical + resources-based) approach. As terminological resource, BabelFy has been used.

Concept linking, which was originally not foreseen in the DoW, but turned out to be necessary for better performance, uses the BabelFy linking services for BabelNet. In this context, also the task of word sense disambiguation has been tackled. For relation extraction and language analysis in general, novel parsing technologies (among them those based on recurrent neural networks) have been developed. The resulting structures are multiple frames in the sense of FrameNet.

The work related to the topic of multimedia concept and event detection addressed primarily the aspect of feature extraction for the description of key frames identified during the video decoding procedure.

All technologies have been evaluated. The evaluation showed that for a considerable share of them, the performance hit the highest expectations.

Abbreviations and Acronyms

API	Application Programming Interface
ASR	Automatic Speech Recognition
BoW	Bag of Words
CEP	Concept Extraction Pipeline
CoNLL	Computational Natural Language Learning
CTM	Compressed Triangle Mesh
CV	Cross Validation
DCNN	Deep Convolutional Neural Network
DNN	Deep Neural Network
DSynt	Deep-Syntactic
EL	Entity Linking
FTB	French Treebank
G2P	Graphic-to-Phoneme
GATE	General Architecture for Text Engineering
GMM	Gaussian Mixture Model
HTTP	Hypertext Transfer Protocol
IDF	Inverse Document Frequency
JSON	Java Script Object Notation
JSON-LD	Java Script Object Notation for Linked Data
LAP	Labeled Attachment Precision
LAR	Labeled Attachment Recall
LAS	Labeled Attachment Score
LM	Language Model
LSTM	Long Short Term Memory
LU	Lexical Unit
ML	Machine Learning
MMM	Multimedia Modeling
NE	Named Entity
NER	Named Entity Recognition
NIF	Net Immerse Format
NP	Nominal Phrase

OOS	Out of Vocabulary
PoS	Part of Speech (tag)
RBF	Radial Basis Function
RDF	Resource Description Framework
REST	Representational State Transfer
RGB	Red Green Blue
SCLITE	Score Lite
SIFT	Scale-invariant feature transform
SKOS	Simple Knowledge Organization System
SRILM	SRI Language Modeling Toolkit
SRL	Semantic Role Labeling
SRT	SubRip Text
SSynt	Surface Syntactic
SURF	Speeded Up Robust Features
SVM	Support vector machine
TF	Term Frequency
UAP	Unlabeled Attachment Precision
UAR	Unlabeled Attachment Recall
UAS	Unlabeled Attachment Score
UC	Use Case
VLAD	Vector of Locally Aggregated Descriptors
W3C	World Wide Web Consortium
WebVTT	Web Video Text Tracks Format
WER	Word Error Rate
WSD	Word Sense Disambiguation

Table of Contents

1 INTRODUCTION

This deliverable reports on the work done in WP2 of the MULTISENSOR project during the third and last year of the project. The main goal of WP2 is to extract knowledge from multimedia documents and encode it in machine-processing formats that facilitate storage and interoperability between MULTISENSOR services.

The current report comprises the following tasks of WP2:

1. T2.2: Named entity extraction workflows
2. T2.3: Concept extraction from text
3. T2.4: Concept linking and relations
4. T2.6: Multimedia concept and event detection
5. T2.7: Machine translation

Additionally, we describe two components that were not foreseen in the DoW but turned out to be important for efficient text processing and rich semantic analysis: a language identification component, and an Entity Linking (EL) module. The latter is reported as part of T2.4, to which it contributes.

All mentioned WP2 tasks contribute to the milestones MS5 of the project (final prototype of the MULTISENSOR system). They correspond to the third year (Y3) activities A2.1 to A2.6, described in the project roadmap D7.1 as shown in Figure 1.

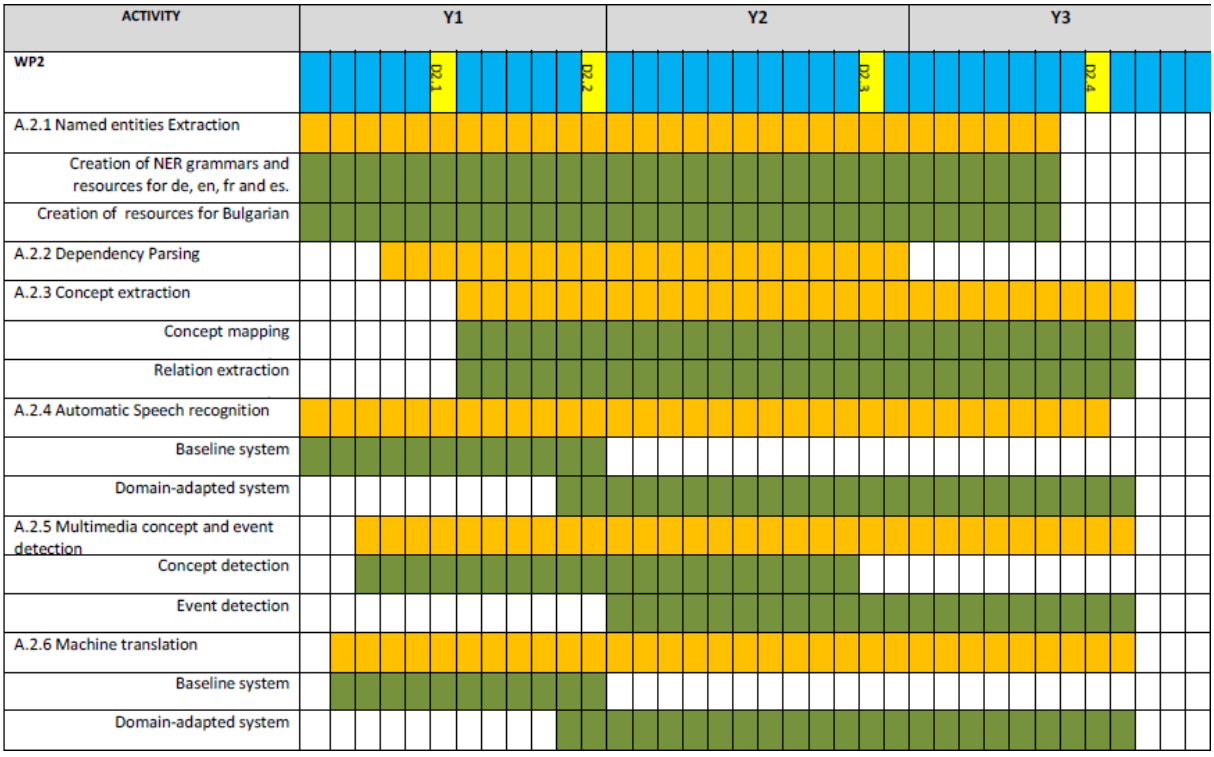

Figure 1: WP2 Roadmap

In this deliverable, we report on each task in a different section. The introductory section gives an overview of the information extraction pipelines and the general architecture of WP2, while in Section 6 some concluding remarks are provided.

1.1 Architecture of the content extraction module

As described in D2.3, there are two main analysis pipelines in WP2: text analysis and video/image analysis. The text analysis pipeline has seen two modifications. First, an Entity Linking (EL) module has been introduced to establish disambiguated links between text fragments in the input documents and entries in a large encyclopaedic and lexicographic database. Second, the concept extraction module now uses the output of the dependency parsing service. For this reason, its execution has been postponed until the end of the Content Extraction Pipeline (CEP). In Figure 2, the CEP is depicted, with the modules addressed in the current deliverable highlighted.

As already reported, all WP2 modules are deployed as REST web services. They communicate with each other via public APIs and exchange JSON messages. The extracted information is encoded as RDF triples and embedded in the JSON messages using JSON-LD. More information can be found in WP7 deliverables (D7.1 to D7.6).

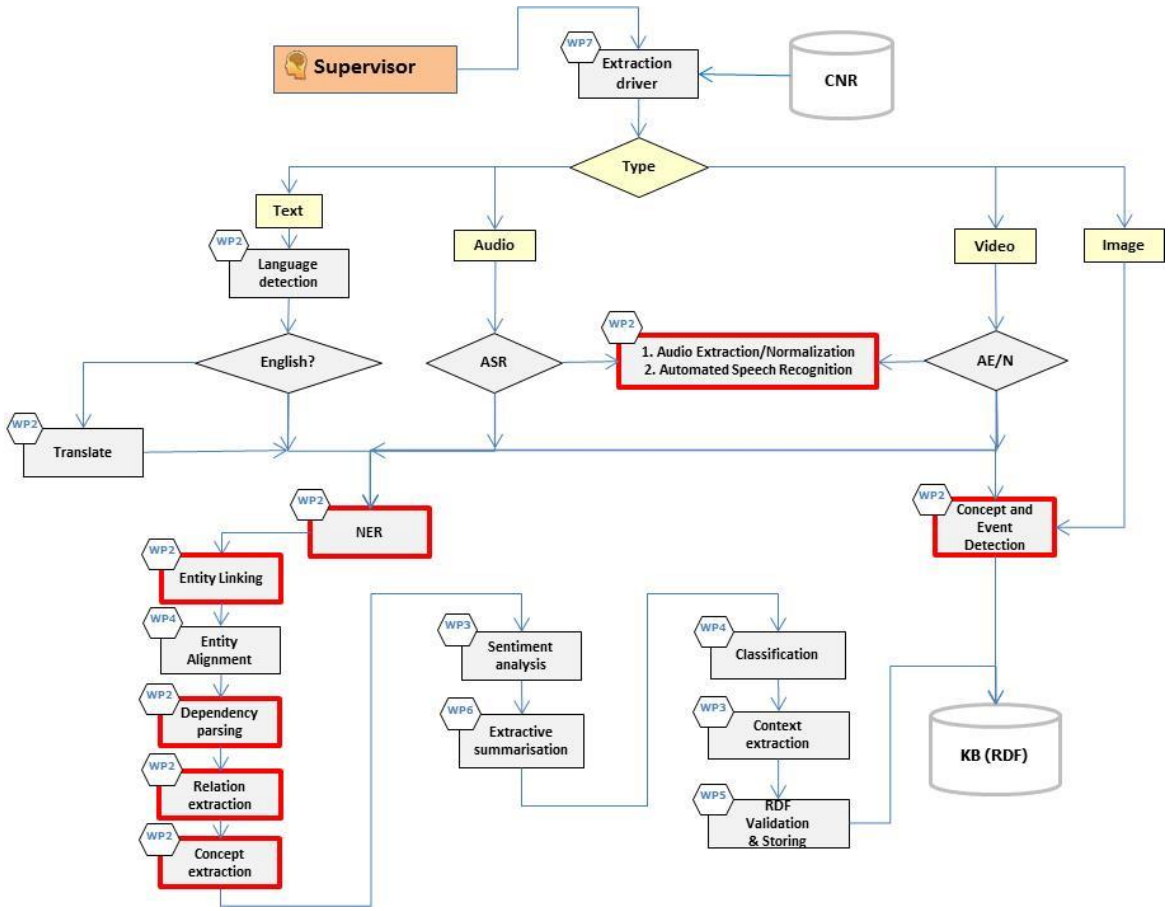

Figure 2: Content Extraction Pipeline architecture with the WP2 modules addressed in the current deliverable highlighted

2 CONCEPT EXTRACTION FROM TEXT

2.1 Work progress in concept extraction task

In D2.3, the first experiments on term extraction using TermRaider[1], which is available in GATE, have been presented. TermRaider bases the identification of terms on the Kyoto score (Bosma and Vossen, 2010), according to a formula that includes the document frequency of the term and the number of different "hyponyms" that the term has ("hyponym" is in this context any term that contains a nested term). TermRaider revealed two limitations. The first limitation is that it is static and processes the full corpus at once. The list of terms that it provides is thus for the full corpus rather than for a single document. The second limitation is that it does not compare the corpus under study with a more general corpus in order to distinguish common terms from corpus-specific terms.

To overcome these limitations, we designed a hybrid system that combines linguistic detection of candidate NPs followed by a statistical scoring and a module based on BabelFy. The scores measure how specific to a domain (or use case) the term is and if the NP (i.e. a multiword construction) can be considered a single term. The results from the different sources are then intersected to produce the final list.

The revised process of concept extraction follows the following steps: 1. Term candidates detection, 2. Statistical feature determination, 3. BabelFy concept identification, and 4. Combining different sources. In what follows, we describe each of these steps in turn.

2.1.1 Term candidates detection

This module, which is language dependent, detects all the NPs of the text that are then considered to candidate terms. The module takes as input tokenized sentences of a document. Tokens are then lemmatized and annotated with POS and syntactic dependencies. To detect NPs, we go over all the nodes of the tree in preorder, finding the head nouns and their dependent elements. A set of rules indicate which nouns and which dependants will form the NP (e.g. relative clauses are not counted as relevant dependents to detect an NP). The system includes a set of rules for each language.

Once detected, each term candidate is expanded with all the subterms (n-grams that compound it) which will also become term candidates.

This step is the only one that is language-dependent. The other steps, even those that use different data for each language, do not need any further language-specific tuning, let alone development, in order to cover other languages.

2.1.2 Statistical feature determination

Each term candidate obtained in the previous step is scored in order to indicate its termhood (that is, how likely it is that the term is a concept) and domain pertinence (to measure if a term is a general domain concept or specific to the domain under study).

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

From the set of different statistical features available in the literature, only some statistical features were chosen for implementation. Based on papers that compare different metrics; see, e.g (Fedorenko et al., 2013) and (Zhang et. al, 2008), we decided to implement the C-Value measure (Frantzi et al., 1998) and the Weirdness metric (Ahmad et al., 1999). The first of them captures the termhood of a candidate term, while the second measures how domain-specific a term candidate is.

The C-value is computed in the following way:

$$C-value(t) = log_2|t|.TF(t) \tag{3.1}$$

if t is not nested into other terms,

$$C-value(t) = log_2|t|.TF(t) - \frac{\sum_{b \in T_t} TF(b)}{|T_t|} \tag{3.2}$$

otherwise.

In (3.2), *TF* is the term frequency, $|t|$ the number of words in the term, $|T_t|$ the number of candidate terms that contain *t*, and *b* candidate terms that contain *t.*

The original Weirdness metric is computed as follows:

$$Weirdness(t) = \frac{TF_{target}(t) \cdot |Corpus_{ref}|}{TF_{ref}(t) \cdot |Corpus_{target}|} \tag{3.3}$$

where *TF* is the term frequency in the target domain or the reference domain and $|Corpus|$ is the size or number of documents in the corpus. One problem of this measure is that it can range from 0 to infinite, which is not desirable. To keep the possible values in a limited range, we change the quotient between probabilities to a quotient between IDF's, so, previous formula (3.3) is transformed to

$$DomWeight(t) = \frac{IDF_{ref}(t)}{IDF_{target}(t)} \tag{3.4}$$

where IDF is the inverse document frequency computed as

$$IDF(t) = log\left(\frac{|Domain|}{TF(t)}\right) \tag{3.5}$$

2.1.3 **BabelFy concept identification**

For the entity linking and word sense disambiguation task, we decided to integrate the BabelFy service (see next section). This service annotates all the terms and named entities found in Wikipedia, but it does not indicate which of them are domain-dependent and does not rank them (this is because BabelFy is a generic domain tool). For this reason, we consider BabelFy to be a source of term candidates, but not a term extraction tool.

2.1.4 **Combining different sources**

At this point of the process, we have two lists of concepts: the list obtained by the statistical pipeline and the list generated by BabelFy. In a first step we filter out the terms generated by the statistical pipeline with a DomWeight below 0.8 or nested terms with a lower C-Value

than the one of the term they belong to which are not found as not nested. The remaining terms are sorted by decreasing C-Value and, when there is a tie, by DomWeight.

In order to combine both lists, we intersect them. The reason to proceed this way is that we observed that both strategies offer a high recall with low precision. As the evaluation shows (see Section 3.2), the combination leads to a considerable increase of precision and a still acceptable loss of recall.

2.1.5 **Implementation**

To compute the *IDF(t)*, we indexed the documents in a Solr index[2], with a field indicating the domain to which they belong. In the current index, we have four groups of documents, a set of 22000 from several domains that constitute the reference corpus, and about 1000 news for each use case.

The use of Solr allows us to have an incremental system, where new documents are indexed and the statistics are continuously updated.

The documents indexed in Solr include the text with all the term candidates in it. To index the term candidates, and in order to allow queries matching the full term (being part of a bigger term or not) or parts of it, we index the term candidates with underscores between the corresponding lemmas. As an example, the term candidate "*real time clocks*" would be indexed as "*real_ _time_ _clock*". Once indexed, we can find terms that are exactly the same term with the query: "*real_ _time_ _clock*". But if we want to know whether this term can be found nested in other terms, we can do it by searching:

"*real_ _time_ _clock_" OR "_real_ _time_ _clock_" OR "_real_ _time_ clock_*".

In this case, the first part of the query would match with "*real time clock synchronization method*" while the third one with "*near real time clock*".

The ideal method to proceed for each new document would be to first index it in Solr and then manage the Solr data. But Solr does not guarantee a real-time update of the query results after a new document has been introduced. This implies that all the cache memories and other intermediate structures used to compute the query must be cleared. For this reason, the service indexes the document after computing the statistics, and when doing so, it adds to the statistics offered by Solr the modification introduced by the document under study.

As we found that the size of the documents is highly variable, we split the documents into groups of 20 sentences (except for the last one that can range between 10 and 30 sentences).

Using Solr and indexing every new document, we can ensure that the response of the system will dynamically adapt to the changes of the domain detecting the emergence of new terms. The code for the concept extraction module is provided at **https://github.com/talnsoftware/concept_extraction**.

[2] **http://lucene.apache.org/solr/**

2.2 Evaluation

To evaluate the concept extraction task, three annotators annotated a text composed of 20 sentences for each of the 3 uses cases of the project (energy policies, household appliances and yoghurt industries). Table 1 indicates the number of terms annotated for each use case and the number of indexed documents (after being split).

In order to evaluate the system and observe the impact of merging the two approaches, we measured separately the performance of the statistical and BabelFy (in more general terms, dictionary) approaches. Then, we measured the performance of the final system. Table 2 shows the precision and recall of the two different approaches and of their merge ("Hybrid System").

Use Case	Domain	Number of documents	Number of indexed splits	Annotated terms
-	Reference Corpus	21994	43308	-
1.1	Household Appliances	1000	2171	123
1.2	Energy Policies	1000	1565	80
2	Yoghurt Industries	1000	2096	118

Table 1: Number of terms annotated for each use case and number of indexed documents

Use Case	Statistical Approach		BabelFy Approach		Hybrid System	
	precision	recall	precision	recall	precision	recall
1	38,1%	93,5%	50,3%	76,4%	65,2%	71,54%
2	28,0%	97,3%	36,2%	74,68%	48,3%	70,9%
3	34,8%	79,5%	46,2%	68,4%	60,9%	57,3%
Avg	33,6%	90,1%	44,2%	73,2%	58,1%	66,6%

Table 2: Precision and recall of different approaches

It can be observed that the hybrid system increases the precision between 14 and 25 percent while the recall decreases between 7 and 24%. To measure whether the increase on precision compensates for the loss of coverage, we computed the F-score, shown in Table 3:

Use Case	Statistical Approach	BabelFy Approach	Hybrid System
1	54,1%	60,7%	68,2%
2	43,5%	48,8%	57,4%
3	48,4%	55,1%	59,1%
Avg	49,0%	55,1%	62,1%

Table 3: F-score of different approaches

Table 3 shows that the F-score of the hybrid system is 7% over the score of the BabelFy approach and 13% above the statistical approach. These results reflect the processing of all terms provided by both tools and only after filtering out the extreme cases. But if we only use the top terms, the precision is higher. We do not implement a threshold to cut the list because only the top N terms are used.

Figure 3 shows how precision and recall evolve as we move down to the list of terms sorted by the score obtained with the statistical tool (BabelFy does not provide any confidence score). Clearly, the scoring puts the most relevant terms at the beginning of the list

increasing the precision by more than 25 points over the average (the first 30 terms maintain a precision over 70%).

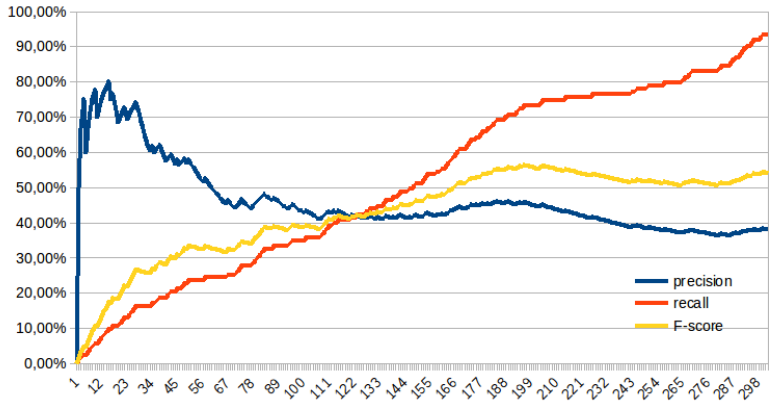

Figure 3: Precision, recall and F-score for annotated texts of UC1.1 using only statistical features

The output of the hybrid system for the same use case has a curve of precision/recall/F-score shown in Figure 4. It illustrates that the first 20 terms keep the precision level at 100%.

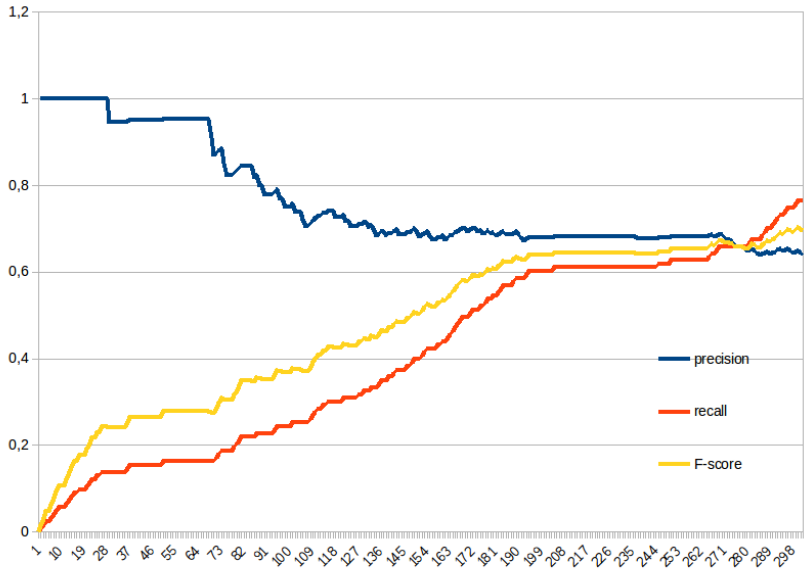

Figure 4: Precision, recall and F-score for annotated texts of UC1.1 using hybrid features

As a conclusion we can state that compared to a baseline system, without scores, and selecting 20 terms at random, we would obtain a precision of 33%, or 44% using BabelFy. Using scores, the precision increases up to 47.7% and when combining both systems, the precision for the 3 uses cases increases to 73.6%, resulting in an overall increase of 40% (achieved to the highest expectation, according to the indicators established in deliverable D1.2 (Self Assessment plan v2)).

3 CONCEPT LINKING AND RELATION EXTRACTION

The focus of the extraction of relations has shifted since D2.3 to finding links between text fragments and entries in an external body of reference, and extending the extraction of relations to support additional languages. While in D2.3 we reported efforts in linking the concepts identified in the concept extraction task to the BabelNet database, the linking has been extended beyond domain-specific concepts to all concepts and NEs covered by BabelNet. Relations are identified by analyzing the functional structure of sentences using dependency parsers, and then operating on the resulting analysis to assign semantic labels to predicative words and their arguments. The functionality described has been implemented and integrated into the Content Extraction Pipeline (CEP) task, enabling the MULTISENSOR prototype to extract n-ary relations involving concepts, NEs and other relations. This functionality is presented in Section 4.1 and an evaluation is described in Section 4.2.

In more specific terms, the concept linking and relation extraction are carried out by a text analysis pipeline that takes as input the textual contents of a document in a given language (see Figure 5 below). This document is first disambiguated, analyzed and represented as a forest of surface-syntactic structures, which are in their turn "transduced" into deep-syntactic structures. Then, if the input language is not English, every lexeme is mapped to the corresponding English lexeme. The English structures are then mapped to semantic structures, enriched with Frames from the FrameNet lexicon (Baker et al., 1998), modeled as RDF triples, and stored in a semantic repository.

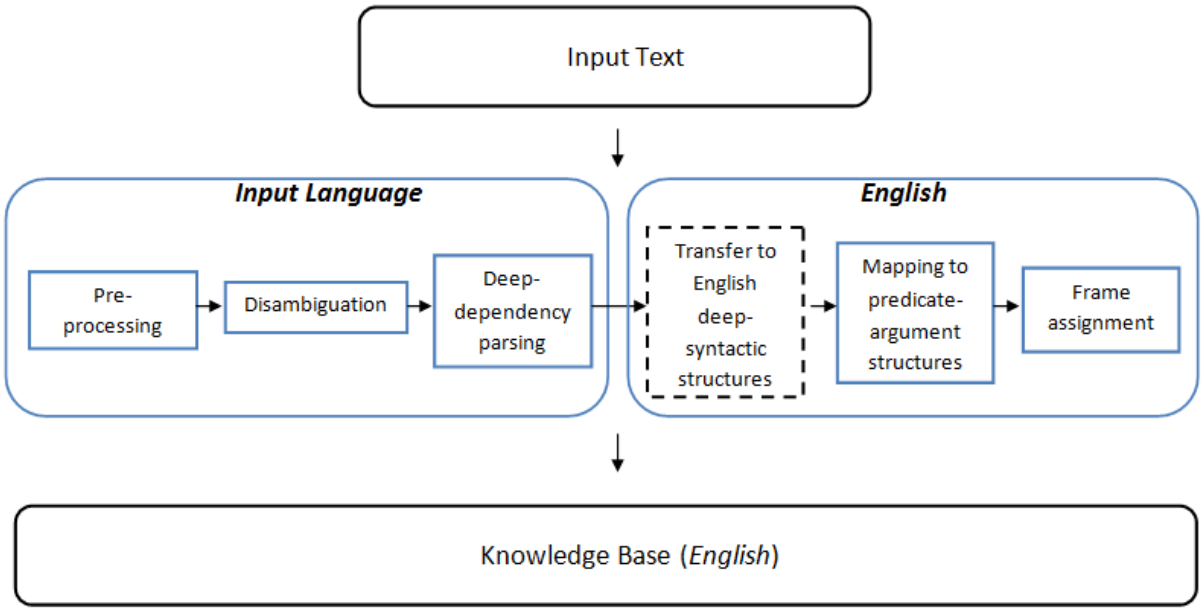

Figure 5: General architecture of concepts linking and relation extraction

3.1 Work progress in concept linking and relation extraction task

3.1.1 Tokenization and disambiguation

Language analysis starts by determining sentence and token boundaries using Bohnet et al. (2013) tools. Rather than addressing tokenization at word level, however, our analysis

pipeline treats each sequence of words referring to a specific entity as an atomic unit of meaning. In doing so, we seek to avoid unnecessary internal analysis of multiword expressions which may not even have a strictly compositional meaning (e.g. “United States of America”), and also to eventually obtain predicate-argument structures in which the arguments are not just words but expressions with an atomic meaning. Thus, all multiword expressions annotated by Babelfy are considered by the following modules as a single token.

3.1.2 **Entity Linking**

The ultimate goal of the relation extraction task is to produce n-ary relations, in which all of the participating entities are semantically defined. Dependency parsers and semantic role labelers are capable of identifying predicative words, disambiguate their senses, classify their meanings into a set of semantic classes describing common relational meanings, and assign semantic roles to their arguments. However, these tools identify arguments as individual words or sequences of words grouped according to their functional relation to the predicate, while we would like to replace text fragments with direct references to their meanings (referred entities, concepts or other relations). Finding the meaning of arbitrary text fragments is addressed by the Entity Linking (EL) and Word Sense Disambiguation (WSD) tools, which detect mentions of entries in an external database or dictionary of senses. While the MULTISENSOR NER implementation detects mentions to certain types of entities, it does not cover concepts and does not produce links to the entities. For this reason, we decided to deploy a new module in the CEP pipeline capable of performing both EL and WSD. While these tasks are not foreseen in the DoW, we consider them to be crucial to produce semantically meaningful relations. After obtaining a license for research purposes, the BabelFy linking service for BabelNet has been used in the module without any limitations such as, e.g., rate limits. The annotations produced by the module are modeled as RDF using NIF, ITS and SKOS vocabularies. Multiword expressions annotated by the module are then interpreted as a single token by the dependency parsing and semantic role labeling modules, so that whenever possible the relations produced by the CEP have as arguments text fragments linked to BabelNet entries.

As already pointed out in D2.3, we have attempted to develop our own graph-based implementation of an EL and WSD component for BabelNet, based on the BabelFy service[3]. However, after the evaluation of this implementation, it has been decided not to integrate it into the prototype because its amelioration beyond BabelFy was considered not feasible within the lifetime of the project. Work on it will continue after MULTISENSOR has finished.

3.1.3 **Dependency parsing**

4.1.3.1 **Surface-syntactic parsing**

Multilingual statistical dependency parsers developed in MULTISENSOR showed cutting-edge results on widely used surface-syntactic datasets. The main contribution has been a new dependency parser (Dyer et al., 2015; Ballesteros et al., 2016) and a control structure for sequence to sequence neural networks that allows for modeling stack like structures. In

[3] The purpose of the implementation was not to avoid licensing issues related to the use of BabelFy, but, rather, to achieve a better performance, using BabelFy as basis.

addition, character-based representations of words were explored, the idea behind which was to use a recurrent neural network to capture morphosyntactic clues, replacing standard look-up based word representations by orthographical representation of words. This implied statistical sharing across word forms that are similar on the surface (Ballesteros et al., 2015b; Ballesteros et al., 2016) and improvement in morphologically rich languages. This new parser is implemented in C++ for performance time purposes, while the rest of the pipeline is implemented in Java. This makes it difficult to integrate it in the architecture of MULTISENSOR[4]. However, the new parser has the potential to be faster (since it runs in a single core), lighter (it only requires 1GB of RAM memory) and more accurate when the same resources are used. The code of the parser can be found at **https://github.com/clab/lstm-parser/tree/char-based**; sample dependency parse is shown in Figure 6.

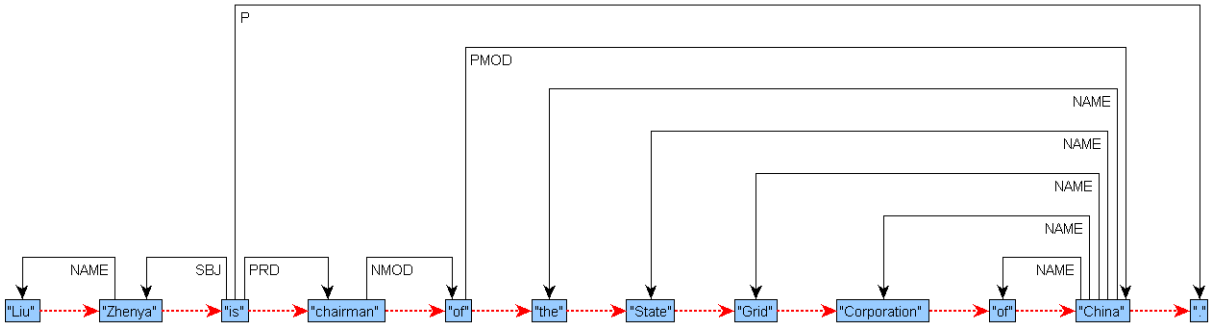

Figure 6: Sample sentence with surface-syntactic annotation

4.1.3.2 **Deep-syntactic parsing**

In order to abstract from language-specific features of the aforementioned dependency parsers, we also aim at structures in which only content-bearing words are present and semantics-oriented relations between them are made explicit. For this, first-in-their-genre multilingual rule-based and statistical deep-syntactic transducers have been developed.

The objective of this kind of transducer is to identify and remove all functional words (auxiliaries, determiners, void prepositions and conjunctions), and to generalize the syntactic dependencies obtained during the previous stage, while adding subcategorization information for syntactic predicates.

In Figure 7, the deep-syntactic structure corresponding to Figure 6 is shown; functional words such as “of” below “chairman” have been removed, and edge labels are oriented towards semantics instead of syntax.

[4] In the CEP, we use the joint lemmatizer, part of speech tagger, morphology tagger and dependency parser of Bohnet et al. (2013) system. This parser follows a transition-based approach with beam search.

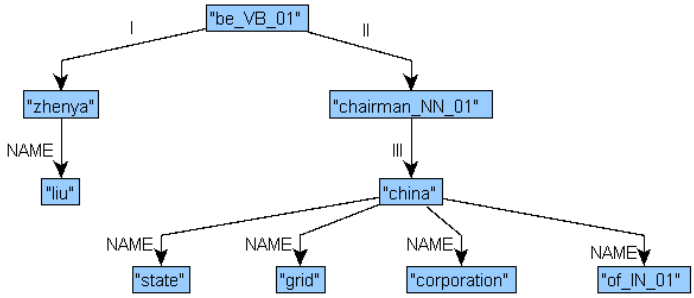

Figure 7: Sample deep-syntactic structure

We offer two options for transducing surface-syntactic dependency trees into deep-syntactic trees: a statistical system and a rule-based system. The statistical transducer (Ballesteros et al., 2014; Ballesteros et al., 2015), trained on parallel SSynt and DSynt corpora (see, for instance, Mille et al. (2013) for Spanish), has been developed for English and Spanish. The rule-based transducer consists of graph-transduction grammars that access language-specific lexicons to remove the void prepositions and conjunctions, when any are available (Mille and Wanner, 2015), and assign predicate-argument edge labels between the remaining words. We have developed rule-based transducers for English, Spanish, German and French. The code for the statistical deep-syntactic transducers is provided at **https://github.com/talnsoftware/deepsyntacticparsing**; the rule-based transducers at **https://github.com/talnsoftware/DSynt_Converter**.

4.1.3.3 **Mapping to abstract representations**

As mentioned in D2.3, we initially planned to use Semafor as a disambiguation tool for both abstract concepts and relations. But the fact that it is not possible to run it concurrently is a major issue when it comes to extracting content from a large amount of documents. Other important limitations are the low reusability of FrameNet structures for the purpose of Natural Language Generation, and the fact that Semafor only operates on English. As a result, we decided to implement an in-house (fast) frame semantics parser that predicts (i) frames according to the FrameNet nomenclature, and (ii) simple predicate-argument relations between words of any part-of-speech (whether they have an associated frame or not), starting from deep-syntactic structures.

For mapping deep-syntactic structures to more abstract linguistic representations, large scale lexical resources are needed. Unfortunately, such resources are only available in English at this point; e.g. PropBank (Kingsbury and Palmer, 2002), FrameNet (Baker et al., 1998), VerbNet (Schuler, 2005) and the mappings between them (SemLink, Palmer (2009)). For this reason, we choose to map all input languages to English. As already pointed out above, after the SSynt-DSynt transduction, all idiosyncratic words are left out, and only meaningful ones are still in the structure. In other words, the parallelism between the deep-syntactic representations of different languages is such that substituting word labels of a language X to English word labels produces a correct English deep-syntactic structure. Using multilingual resources such as BabelNet, it is possible to obtain the translations of these words into English. The analysis pipeline for the different languages therefore requires the compilation of lexicons. For English, they have been automatically obtained from existing resources (PropBank and NomBank in particular); for other languages, we developed a method for manual compilation, explained in the next subsection.

Once this is done, the combination of the subcategorization information in the deep-syntactic structure and SemLink allows for obtaining Frame annotations on top of connected predicate-argument structures. The latter follow the meaning-text approach (Melčuk, 1988), with the addition of a subset of relations such as Location, Time, etc. that facilitate the further processing. During this step, shared argumental positions are made explicit and idiosyncratic structuring such as the representation of raising and control verbs are generalized. Figure 8 is the most abstract representation of the deep-syntactic structure shown in Figure 7. “Chairman” has been associated to the generic frame “Leadership”, which directly connects “Zhenya” and the company name, the first and third arguments of the frame respectively, according to the NomBank role description. The code of our frame semantics parser can be found at
https://github.com/talnsoftware/FrameSemantics_parser.

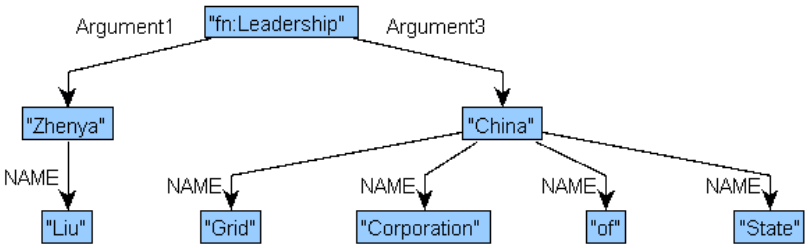

Figure 8: Sample semantic structure with assigned Frames

4.1.3.4 **Experiments with multilingual frames assignment**

In order to design a multilingual generation pipeline, it is necessary to access the lexical knowledge encoded in each language. Thus, we need to create lexical resources for each language covered by the system. These lexicons must not only be monolingual, but also be somehow linked to each other in order to allow for the mapping from English to each of those languages. Given that BabelNet senses annotated during the analysis stage are language-independent, we use them as the cross-linguistic link needed for the multilingual generation pipeline. Below, we detail the creation procedure and structure of the language-specific lexicons used to go from predicate-argument structures and BabelNet synsets to each language, tested using as basis two texts in each language -Spanish, French and German- of the energy policies domain (around 3500 words).

The compilation of the language-specific lexicons was done in different stages. Given that word sense ambiguity is a problem inherent to any language, it was necessary first to disambiguate and recognize the right sense of a lexical unit before assigning any specific BabelNet id to it. The WSD tool BabelFy, which is connected to BabelNet, was used to deal with this problem: using the API offered to remotely access the service, the selected texts were passed through BabelFy. As output of this step, a list of non-duplicated BabelNet ids (1013 items in total) was obtained, which served as the basis for testing the lexicons. This list was locally enriched with the word form linked to each id in each language, in order to facilitate and accelerate the manual compilation of units.

Using the described list as basis, for each LU, its PoS (which refers to a most general entry), its lemma, its BabelNet id and its government pattern (the elements required by the unit, i.e., its subcategorization frame) are stored. Within the government pattern, the information collected for each argument includes its part of speech, the preposition introducing it (if it is

required by the described LU) and the corresponding case. As example, we show below the entry for the same specific BabelNet id in German (a language with case) and in Spanish:

SPANISH

```
"contar_VV_01":_verb_ {
lemma = "contar"
bn = bn:00091011v
gp = {
   I = {dpos = "N"}
   II = {dpos = “N”}
   III = {dpos = "N" prep = "a"}}}
```

GERMAN

```
"sagen_VV_01":_verb_ {
lemma = "sagen"
bn = bn:00091011v
gp = {
            I = {dpos = "N" case = "nom"}
   II = {dpos = "N" case = "acc"}
   III = {dpos = "N" case = "dat"}}}
```

So, from the English structure, the system turns to the lexicons to obtain information about the specific characteristics of the sentences to be generated in each language. If no specific information is added (as in the second argument for Spanish), the system concludes that there are no restrictions with respect to the argument in question (e.g. the second argument in Spanish could be a noun, but also a subordinated verb). Thus, the four compiled parallel language-specific lexicons cater in a direct way to the multilingual generation pipeline, allowing the mapping from English to any of the other languages included. Potentially, the mapping could be even done not only from English to other language, but from any other language included in the system to each other.

3.2 Evaluation

3.2.1 Evaluation of the dependency parsers

The established indicators according to D1.2 are shown in Figure 9:

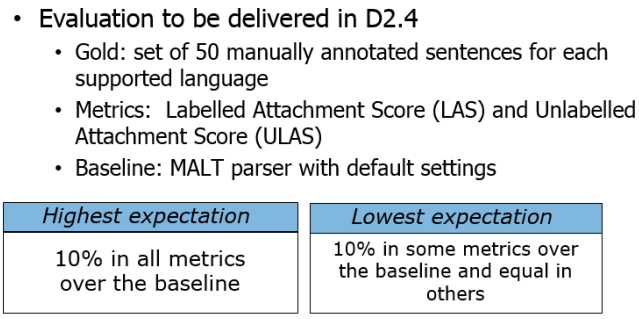

Figure 9: Indicators established in self-assessment plan for dependency parsing

Table 4 presents results of Maltparser (Nivre et al., 2007) in default settings and the parser of Bohnet et al. (2013) which is the parser implemented in the Multisensor pipeline, and Ballesteros et al. (2016) parser with its best performing configuration. We present results (un/labeled attachment score) in reference benchmarks (CoNLL 2009 treebanks (Hajič et al., 2009) and French FTB (Candito et al., 2010)) for the languages used in the project. These annotations all stem from manually annotated data. Instead of using a subset of 50 sentences as stated in D1.2, we used the whole evaluation sets in order to make the comparison with other systems possible. As shown in Table 4, the parser used (Bohnet et al.,

2013)[5] and the parser developed within the project (Ballesteros et al., 2016) overcome the results of MaltParser, which is our baseline, for all languages. Both parsers comply with the highest expectation of the project, which is error reduction in more than 10% in both metrics: LAS and UAS. The error reduction is the normal way of measuring the improvement in dependency parser. Note that for English the baseline already achieves ~90% UAS, an increase of 10% would mean to reach an almost perfect accuracy which is not possible even in manual annotation.

For English, both Ballesteros et al. (2016) and Bohnet et al. (2013) improve the baseline by more than 3 points, which corresponds to 37.5% error reduction for UAS. For German, the improvement is higher, especially in the case of Bohnet et al. (2013) parser (+8 points, -49% err. rate), which uses explicit morphological features. Ballesteros et al. (2016) (+7, -43% err. rate) parser achieves a very competitive performance without explicit features. For Spanish, we have a similar picture, with improvements of 6 points (-42% err. rate) for Bohnet et al. (2013) and 5 points (-34% err. rate) for Ballesteros et al. (2016). Finally, for French, Ballesteros et al. (2016) uses the same set of features, including morphology, and improves the baseline by more than 4 points (-25% err. rate), while Bohnet et al. (2013) improves the baseline by almost 4 points (-22% err. rate).

	English	German	Spanish	French
	UAS / LAS	**UAS / LAS**	**UAS / LAS**	**UAS / LAS**
MaltParser Default Settings (2007)	88.64 / 86.0	83.1 / 80.7	86.6 / 82.4	82.5 / 78.0
Bohnet et al. (2013) (integrated)	92.9 / 90.6	91.4 / 89.4	92.2 / 89.6	86.4 / 82.6
Ballesteros et al. (2016)	92.3/89.9	90.3 / 88.2*	91.1/88.0*	86.8 /82.7

Table 4: Surface dependency parsing results

Results marked by '*' imply that the results for Ballesteros et al. (2016) parser for German and Spanish were obtained without explicit morphological features, while the French numbers were obtained including morphological features. The English treebank does not have morphological features. The results for Bohnet et al. (2013) parser included morphological features for all languages, if available.

In terms of speed performance, Ballesteros et al. (2016) parser is 4 times faster than Bohnet et al. (2013) parser, since it uses a greedy decoding strategy, being trained to minimize cross-entropy relative to a distribution of gold-standard sequences (obtained by transforming labeled syntactic trees using a manually defined procedure). At test time, the parser makes greedy decisions according to the learned model. However, Bohnet et al. (2013) parser uses beam search, which implies the need to explore more paths. This makes it slower and with higher memory requirements (1GB for Ballesteros et al. (2016) vs. 32 GB for Bohnet et al. (2013)).

[5] https://code.google.com/p/mate-tools/wiki/ParserAndModels

The surface syntactic parsers developed in the framework of MULTISENSOR achieve results that go beyond our highest expectations; not only did we improve the baseline parser by more than 10% (in terms of error reduction), but we also obtained results very close to the best results known to date on reference evaluation datasets. In addition, our tools perform faster and with a significantly reduced amount of memory compared to parsers with a similar performance.

3.2.2 Evaluation of relation extraction

The indicators established in D1.2 with respect to relation extraction are summarized in Figure 10.

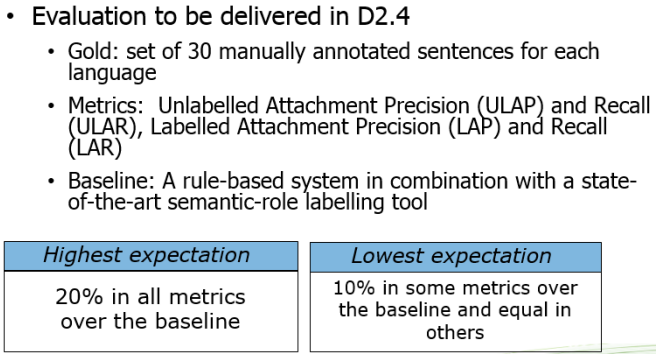

Figure 10: Indicators established in self-assessment plan for relation extraction

The gold standard reported in D2.3 has been used to evaluate the output of the English relation extraction module. This corpus contains manual annotations of NEs, linguistic predicates and their arguments, and FrameNet frames and roles assigned to predicates and their arguments. Predicate-argument structures in the gold are equivalent to the n-ary relations produced by the module, except that the MULTISENSOR module does not assign FrameNet roles to arguments, but, instead, produces PropBank-style roles, i.e 'Arg1', 'Arg2', 'Arg3', etc. Arguments that could not be assigned a PropBank role are assigned generic roles labeled *non-core*, or *elaboration*, when none of the two elements can be said to be semantically subordinated to the other. The obtained outputs are comparable to state-of-the art SRL tools such as (Björkelund et al., 2010).

The evaluation has been conducted on a subset of the gold consisting of 10 sentences belonging to each use case, up to a total of 30 sentences. The assignment of frames to predicates and the detection of predicate arguments have been evaluated separately using the same gold standard, but different baselines. The following criteria have been followed in the evaluation of the relation extraction component:

1. Nominal groups marked as NEs or concepts that have a non-compositional meaning are interpreted as a single word. Any internal analysis of these expressions produced by the evaluated tools is ignored rather than being counted as correct or incorrect predictions.
2. Prepositions and conjunctions that bear their own meaning are annotated in the gold as non-core dependents of a predicate, instead of considering them as full predicates. Since it is not clear whether these prepositions indicate relations of their

own, non-core dependents are considered correct predictions, regardless of being annotated with a frame or just as non-core arguments.

3. We have interpreted Elaboration as a bidirectional relation. Thus, we have counted as correct those cases, in which the elements and relation coincide, ignoring the direction of the relation.
4. The labels assigned to relative pronouns by MATE Tools have not been considered, given that they only duplicate the relation assigned to the antecedent of the pronoun. Counting them in the evaluation would produce unnecessary false positives. Relative pronouns with integrated antecedent have been considered as an exception, given that they do not duplicate the relation.
5. Tokenization differences were found between systems in some specific cases (e.g. 71/100, $513 million), for which some sentences had to be manually adjusted in order to facilitate the comparison.

We performed two types of evaluation for the relation extraction: unlabeled precision and recall (Table 5), and partially labeled precision and recall (Table 6). The former is common for the evaluation of relation prediction. The reason for not using all the labels (i.e. for using "partially labeled" instead of "fully labeled") is that in MULTISENSOR, the information needed for selecting the content once the output of the Content Extraction Pipeline is stored in the knowledge base is argument vs. non-argument relations. In other words, we do not use the types of argument relations (first argument, second argument, etc.), but, rather, focus on the question whether an element is an argument of a predicate or not.

The results in Tables 5 and 6 show the precision and recall of the MULTISENSOR relation extraction component, comparing them to the SRL MATE figures. Since the MATE tools (our baseline), only assigns roles to verbal and nominal predicates[6], the results are shown separately for each PoS type of the predicates, such that a meaningful comparison can be made.

PoS	Relevant elements	MULTISENSOR		MATE TOOLS		IMPROVEMENT	
		Prec. (%)	Recall (%)	Prec. (%)	Recall (%)	Prec. (%)	Recall (%)
all	528	74.40	71.02	60.79	37.88	+22.39	+87.49
verbs	191	75.00	78.53	65.48	67.54	**+14.54**	**+16.27**
nouns	182	70.75	57.14	53.79	39.01	**+31.53**	**+46.47**
adj/num	58	71.43	68.97	-	-	-	-
adverbs	13	66.67	76.92	-	-	-	-
conjunctions	74	91.67	89.19	-	-	-	-
possessives	10	35.71	50.00	-	-	-	-

Table 5: Semantic Role Labeling - Unlabeled Attachment Scores

[6] Participles, even if they are acting as adjectives, are also considered verbal predicates.

PoS	Relevant elements	MULTISENSOR		MATE TOOLS		IMPROVEMENT	
		Prec. (%)	Recall (%)	Prec. (%)	Recall (%)	Prec. (%)	Recall (%)
all	528	66.27	63.25	53.19	33.14	+24.59	+90.86
verbs	191	74.50	78.01	63.45	65.45	**+17.42**	**+19.19**
nouns	182	44.22	35.71	37.88	27.47	**+17.27**	**+30.00**
adj/num	58	71.43	68.96	-	-		
adverbs	13	60.00	69.23	-	-		
conjunctions	74	91.67	89.18	-	-		
possessives	10	35.71	50.00	-	-		

Table 6: Semantic Role Labeling – (partially) Labeled Attachment Scores

As shown in Tables 5 and 6, for verbs and nouns, the CEP performs better than MATE Tools, and the improvement is close to our higher expectations (20% increase of the performance). The improvement is even higher than expected for the recall of dependents of nouns, which we believe is mainly due to the fact that, in the same way that MATE Tools does not predict arguments of adjectives, adverbs, conjunctions, prepositions, etc., it cannot predict all nominal dependents.

MATE Tools is trained on the PropBank annotation: it assigns predicate-argument and non-core dependents to the different predicates. Arguments in the CEP are annotated in the same fashion as in MATE Tools (as an ordered list of participants: 'Arg1', 'Arg2', 'Arg3', etc.), but it is unclear when non-core dependents are predicted, since not all of them are annotated in the original PropBank annotation. Nouns have numerous non-core dependents, and a tool that does not predict them all will inevitably have a low recall, as it is the case for MATE Tools.

For other languages, relation extraction is evaluated as deep-syntactic parsing (see Section 4.1.3). In order to evaluate the relation extraction in German and French, 51 sentences have been manually annotated with deep-syntax in each language (942 and 664 words respectively), and compared to the annotation produced by our rule-based transducers[7]. For Spanish, we annotated a gold standard evaluation set of 258 sentences (5641 words); we also provide evaluations for English (1299 semi-supervised annotated sentences for evaluation, 42480 words). For these last two languages, statistical transducers have been developed in addition to rule-based ones. Two aspects are being evaluated:

- Hypernode identification evaluation:
  - $F1h(F1h) = \frac{2ph*rh}{(ph+rh)}$, with 'ph' as the number of correctly predicted nodes divided by the total number of predicted hypernodes, and 'rh' the number of correctly predicted hypernodes divided by the number of hypernodes in the gold standard.

[7] Due to the lack of good quality training material, it has not been possible to develop statistical transducers for German and French; we only report number for rule-based transducers.

- 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.

Table 7 shows the evaluation results for the SSynt-DSynt transitions only, in order not to take into account errors produced by the surface-syntactic parsers.

	Spanish (ML)	English (ML)	Spanish (RB)	English (RB)	German (RB)	French (RB)
F1h	99.51	98.88	97.31	98.12	97.71	97.90
LAP	91.07	90.63	79.57	86.97	89.01	82.12
UAP	98.32	93.70	88.95	90.77	92.72	90.75
LAR	90.57	91.02	83.25	89.08	86.60	83.86
UAR	97.78	94.11	93.07	92.97	90.21	92.68

Table 7: Evaluation results of hypernode detection and attachment and labeling (%; ML: Machine Learning; RB: Rule-based)

As mentioned above, for Spanish and English, we have both rule-based and statistical transducers; we used the rule-based systems as a baseline for the statistical ones. Table 8 shows the significant improvements obtained compared to the in-house baselines, by following the approach of surface-syntactic parsing, making explicit the error reduction rate for each measure as reported in Table 8.

	Improvement Spanish	Improvement English
F1h	81.78	40.43
LAP	53.66	28.09
UAP	84.80	37.74
LAR	43.70	17.77
UAR	67.97	16.22

Table 8: Error reduction rate (%) between baseline and ML system

3.2.3 **Evaluation of frames**

Initially, we did not plan to implement our own disambiguation tool, but it turned out to be necessary for the CEP to be able to process a large amount of documents in a reasonable

time. We implemented a rule-based pipeline that not only assigns predicate-argument relations to the different content words of a sentence (see previous section), but also uses existing resources such as SemLink and the FrameNet lexicon in order to predict frames associated to these words, in a similar fashion as FRED (Presutti et al., 2012) does.

The baseline used for frame-detection is Semafor, a FrameNet-based semantic role labeler employed in previous versions of the relation extraction module and described in detail in D2.3. In this particular evaluation, we have ignored linguistic predicates in the gold which have no frame annotated because none matched their meaning. Predicates annotated with frames but not listed in FrameNet as lexicalizations of the corresponding frame are also excluded, as all systems used FrameNet index as the basis for deciding frame annotations. This affects, for instance, many quantities, which are not listed as lexical units of the *Cardinal_number* frame.

Frames assigned to predicates without arguments have been included in the evaluation as Semafor produces such annotations. However, these frames cannot be considered as true relations due to the lack of any participants. For this reason, we have evaluated them separately. Following the approach adopted in the evaluation of the PIKES system (Corcoglioniti et al., 2016), we have considered three possible values in the match between frames: i) total match, which is counted as 1, ii) partial match (the frames are not the same, but are very related between each other), counted as 0.8; and iii) mismatch.

	Semafor baseline		**MULTISENSOR relation extraction**		**Improvement**	
	Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
Predicates with args	76.85	72.14	71.64	64.69	**-6.78**	**-10.33**
Predicates without args	74.76	66.81	77.84	61.28	**+4.12**	**-8.28**
All	76.46	71.11	72.71	64.03	**-4.9**	**-9.96**

Table 9: Results of evaluation of frame detection

The results of the evaluation are displayed in Table 9, which provides precision and recall figures for both Semafor and the relation extraction system developed within MULTISENSOR. Figures are given for predicates with arguments only (top row), predicates with no arguments only (middle row), and all predicates (bottom row). Table 9 shows that the results do not get to the level of Semafor at this point, which was expected for a system that is not trained on annotated data. Although the numbers are not directly comparable because they were obtained from different reference datasets, our results are in line with those reported by (Presutti et al., 2012) for Boxer, with slightly lower precision but higher recall (75.32% precision and 57.52% recall for Boxer vs. 72.71% precision and 64.03% recall for MULTISENSOR's CEP). In order to assess the speed of our system and compare it with Boxer and Semafor, we used the same dataset as in (Presutti et al., 2012), that is, a gold standard annotation of 1214 sentences with frames. Our surface-syntactic parser needed about 1m 20s to process the whole file, and the relation extraction pipeline another 1m 22s,

which is about 2m 42s, very close to the 2m 45s reported for Boxer. On the same dataset, Semafor has been reported to run in 20m 14s.

3.3 Adaptability

In the domain of syntactic parsing and machine learning, we have conducted research of character-based representation of words with bidirectional LSTMs (recurrent neural networks). The character-based representations are a way of overcoming the out-of-vocabulary (OOV) problem; without any additional resources, they enable the parser to substantially improve the performance when OOV rates are high, since they allow it calculate vector representations for words that the machine learning model has never seen during training (out of domain, mainly). This implies that the machine learning model will be able to handle (and classify) new words without using additional resources.

4 MULTIMEDIA CONCEPT AND EVENT DETECTION

This section presents the advanced techniques implemented in multimedia concept and event detection, which involves the detection of a set of predefined concepts/events in multimedia files. In the work of this deliverable, concept/event detection on video files only is considered. The main steps that comprise the procedure are the following:

- Video decoding: in this step, representative key frames are extracted from the video files.
- Feature extraction: this step refers to the extraction of features that visually describe the key frames.
- Classification: this step involves the training of models, in order to classify videos to the set of predefined concepts/ events.

In the current deliverable, no progress has been made with respect to the video decoding step. What has been modified significantly is the feature extraction process. Specifically, a different type of visual features, based on Deep Neural Networks (DNNs), was extracted. These features, along with an overview of the progress regarding the feature extraction procedure in the MULTISENSOR multimedia concept and event detection framework, are described in Section 5.1. Finally, it should be noted that a description of the state-of-the-art techniques used in the video decoding, feature extraction, and classification steps has been provided in previous deliverable D2.2 (see Section 7.1 – D2.2). Therefore, no further details will be provided in the current deliverable.

4.1 Work progress in multimedia concept and event detection task

Since the problem tackled in this deliverable focuses on concept and event detection in video files and the feature extraction step involves methods being applied to images, it easily follows that video decoding is an indispensable step. In a nutshell, videos are segmented into shots and a representative key frame is selected from each shot.

Feature extraction is the phase where we try to describe the visual information of multimedia efficiently. Before the advent of Deep Neural Networks as an optimal method to calculate visual features, the literature focused on techniques extracting two types of descriptors, namely global and local ones. The global descriptors leverage the global features of an image, whereas local descriptors are computed on sampled points or regions. In addition, while extracting local descriptors, a vocabulary of “visual words” is constructed through the application of a suitable clustering algorithm. With the use of this “visual words” vocabulary, the local descriptors are transformed into a “Bag-of-Words” (BoW) representation (Qiu, 2002) and as a result, a global descriptor that provides a general impression of visual data is produced.

In the first version of the multimedia concept detection framework (see Section 7 – D2.2), we relied on the broadly used SIFT (Lowe, 2004) and SURF (Bay et al., 2008) descriptors and their variations (RGB-SIFT, opponent-SIFT, RGB-SURF, opponent-SURF). The visual word assignment was made using an alternative of BoW named VLAD (Vector of Locally Aggregated Descriptors) (Jegou et al., 2010). In the next version of the multimedia concept detection framework (see Section 6 – D2.3), only SIFT features were extracted and BoW replaced VLAD as a vector aggregation method. Based on the SIFT features, we also

calculated RootSIFT (Arandjelović and Zisserman, 2012) features. In addition, we tried to pre-process the images before feature extraction using hierarchical saliency detection, a technique that aims to find the most significant information of the image (Yan et al., 2013).

In the current version of our multimedia concept and event detection module, we decided to rely on deep convolutional neural networks (DCNNs), one of the most successful and widely used forms of deep networks, in order to learn features directly from the raw key frame pixels. In general, deep learning techniques offer a compelling alternative to traditional architectures for solving problems in computer vision (due to their ability to automatically learn problem-specific features) and therefore, there is a trend to re-examine every computer vision problem from a deep learning perspective (Srinivas et al., 2016). A crucial advantage of DCNNs is the fact that they consist of many layers of feature extractors, something that gives them a more sophisticated structure than hand-crafted representations.

DCNNs can be used either as video key frame feature extractors (see Figure 11, upper part), where the output of a hidden layer of the pre-trained DCNN is used as a global key frame representation (Markatopoulou et al., 2015), or as standalone classifiers (see Figure 11, lower part), where keyframes are passed through a pre-trained DCNN that performs the final class label prediction directly. Several DCNN software libraries are available in the literature, e.g., Caffe (Jia et al., 2014), MatConvNet (Vedaldi and Lenc, 2015), and different DCNN architectures have been proposed, e.g., CaffeNet (Krizhevsky et al., 2012), GoogLeNet (Szegedy et al., 2015), VGG ConvNet (Simonyan and Zisserman, 2014), etc.

In our framework, we extract the novel DCNN features on the key frames of the video dataset, based on the pipeline depicted in the upper part of Figure 11, meaning that we do not use DCNNs in order to train classification models. The DCNNs we used were already trained using the Caffe[8] tool in the work of (Markatopoulou et al., 2016). The network was trained according to the 22-layer GoogLeNet[9] architecture on the ImageNet "fall" 2011 dataset for 5055 categories (Russakovsky et al., 2015). The output of the second last fully connected layer of the second auxiliary classifier was used as a global image representation with a dimensionality of 1024.

After extracting the DCNN-based features, a classification model was trained for each concept and event separately. We chose to work with the well-known Support Vector Machines (SVM) classification algorithm. In order to deal with the linearly non-separable features, we employed the radial basis function (RBF) kernel. Moreover, class weights were adjusted to be inversely proportional to their frequencies, because it is important to train classifiers capable of classifying even the least frequent classes. The SVM implementation we used is the one from scikit-learn[10], a machine learning Python library. We have to note that even though we deal with a video concept/event detection problem, training is made using an image (key frame) set. In the test/prediction phase, in order to get the detected concepts/events from an unknown video, we firstly predict them on its detected key frames. To aggregate the predictions and get the final detected video concepts/events we use the

[8] http://caffe.berkeleyvision.org/

[9] https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet

[10] http://scikit-learn.org/stable/

following simple rule: "If a concept/event is detected in at least one of a video's representative frames, then that video contains the specific concept/event".

Finally, we would like to note that apart from the visual features, we also attempted to extract textual features in order to supplement our framework with an additional modality. The idea was to use the MULTISENSOR Automatic Speech Recognition (ASR) system in order to transcribe the audio in the video files and extract features from the resulting text. However, we noticed that the speech in the video files was not particularly relevant to the concepts/event we aimed to detect. As a result, these textual features would not be of any use to our experiments. Therefore, we decided to focus solely on DCNN-based visual features in our concept and event detection framework.

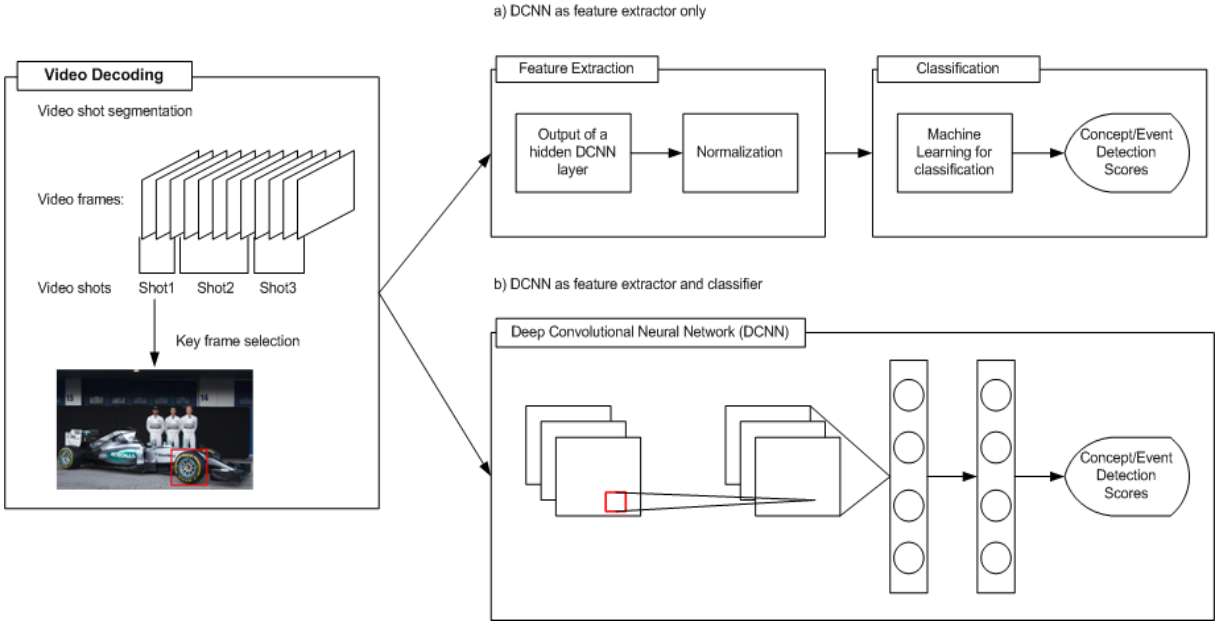

Figure 14: Video concept/event detection pipelines with DCNN-based features

4.2 Multimedia concept and event detection module

The code for the experiments conducted within MULTISENSOR for the multimedia concept and event detection task has been developed in Python, version 3.5.1, 64-bit, (the experiments are presented in Section 5.3). It makes use of many external packages such as numpy and sklearn. The dataset used in these experiments contains videos, which are categorized into nine classes that represent concepts/events. One video may be relevant to zero or more classes. Features were already extracted using DCNNs and this code expects them as an input. Moreover, it expects as input a video-to-shot mapping and the ground truth shot annotations, due to the fact that the feature extraction and the training procedures are executed on shots, while evaluation is conducted on video level. After the evaluation process (described in Section 5.3.3) is completed, values for the accuracy and F-score measures per concept are reported. The code developed within MULTISENSOR for the multimedia concept and event detection module is available at: **https://github.com/MKLab-ITI/multisensor-concept-event-detection**.

4.3 Evaluation

4.3.1 Concept and event selection for MULTISENSOR

Multimedia concept and event detection is applicable only for two MULTISENSOR use cases, namely "Journalism use case scenario" and "Commercial media monitoring use case scenario" (see Section 6.1.1 – D2.3). In order to select concepts/events for our experiments, we followed the approach of visually inspecting exemplary videos provided by the user partners. While in previous deliverables D2.2 and D2.3 we utilized concepts for both use cases, in the current deliverable we focused only on the "Journalism use case scenario", as the visual inspection of exemplary videos showed that it was plausible to define/select interesting concepts and events only for this use case.

It is also important to note that in our framework, any entity a classification algorithm can detect is regarded as either a "concept" or an "event". However, the definitions of these two words are very ambiguous. In most cases, entities describing objects are supposed to be "concepts", while anything that contains an action or something that lasts for a long time are named as "events". Still, all entities cannot be classified exclusively to one of these words. For example, we cannot discriminate whether a "fire" is considered as a concept or as an event. The nine concepts/events selected for the "Journalism use case scenario" that were used in the experiments are displayed in Table 10.

ID	Concept/Event
001	Outdoor factory smoke
002	Wind turbine
003	Solar panel
004	Lattice tower
005	Construction workers
006	People protesting
007	Speaking to camera
008	Fire
009	Airplane flying

Table 10: Selected concepts/events for the "Journalism" use case

4.3.2 Dataset creation

The videos collected for the development of our concept/event detection models are extracts from news reports in order to comply with the "Journalism use case scenario". Most of them were provided by the Deutsche Welle (DW) data repository. The rest were collected from Youtube and include reports from well-known news agencies (e.g. BBC). The videos, as well as their extracted keyframes were manually annotated. It easily follows that any video or key frame can contain zero, one or more of the nine concept/events presented above. Based on the aforementioned procedure, a dataset of 106 videos was formed. The dataset,

along with the ground truth video annotations are available at: http://mklab.iti.gr/files/Event_Detection_Dataset_MS.rar

4.3.3 Evaluation metrics – Experimental setup

In Table 11, we report the positive and negative video examples in the dataset for each concept/event.

ID	001	002	003	004	005	006	007	008	009
Positive percentage	16%	28%	23%	32%	27%	18%	79%	18%	8%
Negative percentage	84%	72%	77%	68%	73%	82%	21%	82%	92%

Table 11: Percentages of positive and negative video examples for each concept/event

We notice that in most concepts/events, there are noticeably fewer positive examples. The exception is the concept with id "007" ("Speaking to camera"), where the positive class is much more frequent. So, in this dataset, there is a rather huge imbalance between the positive and negative classes. In such cases, a classifier categorizing all data to the most frequent class will have a very high accuracy score. Therefore, in this problem, accuracy is not the most suitable metric for evaluating the classification performance for each concept/event. Instead, we utilized three common IR metrics, namely precision, recall and F-score. We would like to note that in Section 5.3.4, only the F-score values are reported for each concept/event, as F-score takes into account precision and recall simultaneously.

Evaluation is made separately for each concept/event detector. The procedure consists of the following steps:

1. Split the dataset into three chunks to setup a 3-fold cross validation (CV) process. The chunks are balanced so that they have almost the same positive and negative examples. At each fold, two chunks are used for training and one for testing purposes.
2. In the training phase, perform a grid search to tune C parameter for SVM. Tuning is made by selecting the optimal C parameter in terms of F-score through an internal 3-fold cross validation procedure on the training set. The search range for the C parameter optimal value is $10^N, -8 \leq N \leq 8$.
3. Using the optimal trained classifier, evaluate its performance on the test set. The average F-score from the three folds is calculated.

Finally, for comparison purposes, we conducted experiments with the previous version of the MULTISENSOR multimedia concept detection framework presented in D2.3, using the aforementioned training and evaluation procedure. In a nutshell, the D2.3 framework relies on SIFT and RootSIFT local descriptors. For each concept/event, a separate SVM classification model is trained for each of these two types of features. Then, a weighted average late fusion strategy is applied, in order to produce the final predictions for each concept/event during the testing phase (for more details, see Section 6 – D2.3).

4.3.4 Evaluation results

In Table 12, we report the classification performance results for the previous, as well as the current version of the MULTISENSOR concept/event detection framework, using local (SIFT + RootSIFT – D2.3) and DCNN (D2.4) features, respectively.

In general, we notice that the models trained on DCNN features achieve a good performance. Considering how demanding the dataset is due to the few available positive examples for each concept/event, a macro-average F-score value of 72.53% for the DCNN-based framework is satisfactory. Nevertheless, there is considerable variation if we look at the F-score values per concept/event independently. As expected, the concept/event with the largest number of positive examples ("Speaking to camera") has the highest and almost perfect F-score value (94.38%). On the other hand, for some concepts/events like "Outdoor factory smoke" and "Construction workers", the DCNN models have a difficulty in categorizing the test set videos. But an F-score well above 60% in seven out of nine concepts/events show that good performing classifiers are produced using this framework.

Compared to the performance of the framework version, in which local features are utilized, it's obvious that DCNN features are much more suitable for this task. The classification models trained on DCNNs yield an extremely higher macro-average F-score (72.53% compared to 35.2% for the local features models). There are two main reasons for this. Firstly, in D2.3 we concluded that the local features are very efficient in cases, where we try to detect objects like logos. The addition of events (e.g. Outdoor factory smoke) revealed the weaknesses of these features and demonstrates the fact that they are not suitable for this multimedia concept/event detection problem. While in D2.3 the evaluation was performed on images, in the current deliverable it was performed on videos, therefore, the problem is different in this aspect. Furthermore, the results show that the DCNN features fit better to our video concept and event detection framework.

Finally, it easily follows that according to the indicators established in deliverable D1.2 (Self Assessment plan v2) for T2.6, the multimedia concept and event detection task has been achieved to the highest expectation, as there is an F-score (considers both precision and recall) performance improvement of over 5%, compared to the baseline system (we consider the local features framework version as the baseline system).

Concept/Event name	**Local Features (SIFT/ RootSIFT)**	**DCNN features**
Outdoor factory smoke	11.11%	55.41%
Wind turbine	46.04%	82.62%
Solar panel	48.33%	86.31%
Lattice tower	54.50%	69.30%
Construction workers	21.16%	58.97%
People protesting	38.21%	71.08%

Speaking to camera	64.74%	94.38%
Fire	6.06%	64.66%
Airplane flying	26.67%	70.00%
Macro-average	**35.20%**	**72.53%**

Table 12: Evaluation results for the two concept/event detection frameworks using local (D2.3) and DCNN (D2.4) features

4.4 Adaptability to other domains

With respect to the adaptability of the multimedia concept and event detection module to other domains, it should be noted that a common problem in video datasets is the lack of sufficient numbers of labeled training examples. Therefore, the training of a deep network from the ground up without over-fitting its parameters is a difficult task (Snoek et al., 2015). To this end, transfer learning is commonly used by taking a network that has been trained on a large-scale dataset from a source domain and fine-tuning its parameters for a dataset coming from a target domain. In other words, transfer learning aims at improving the learning in the target domain by utilizing the knowledge present in the source domain, without considering improvements to the learning tasks of the source domain.

There have been several DCNN-related research studies focusing on the development of techniques for efficiently transferring knowledge to new target datasets. Typically, in transfer learning we start with a DCNN trained in the source domain, replace its classification layer with a new one and train it towards the target domain (Girshick et al., 2014; Yosinski et al., 2014). In order to perform fine-tuning, we begin with the parameter weights of the source domain DCNN and the aim is to modify them, so that the network can be efficiently adjusted to the target domain.

In conclusion, based on all the aforementioned, the adaptation of the MULTISENSOR multimedia concept and event detection module, presented in the current deliverable, to other domains is a feasible task through the use of transfer learning and fine-tuning.

5 CONCLUSIONS

This document describes the final achievements of WP2 in the areas of concept extraction, concept linking and relation extraction and multimedia concept and event detection.

With respect to **concept extraction**, systems based on linguistic features are better in order to find very rare terms but they become language and domain dependent. The same applies to the systems that use gazetteers or dictionaries; a dictionary is needed for each language. Nowadays there exist resources like BabelNet, a multilingual semantic network and ontology. BabelNet has been generated automatically by linking Wikipedia which is kept up to date by an active crowd of volunteers. For this reason, Wikipedia can be considered up-to-date and covers many domains, also very specific ones. But as a general purpose tool, BabelNet does not indicate which of the terms are the domain specific ones. Statistical tools provide many term candidates that are domain specific and common enough to be considered terms but maybe semantically soundless. Both approaches offer a high recall at the expense of low precision because each of them adds its own noise. When combining the two techniques we can increase the precision but losing some recall. The decrease on recall is overcompensated by the increase on precision improving the F-score. This increase is more evident when we concentrate on terms with higher score. The use of an index like Solr to keep the corpus data allows the creation of a dynamic system that can be updated with upcoming news, making the response dynamic when new concepts appear in a domain.

With respect to **concept linking and relation**, we presented a multilingual analysis pipeline that is able to produce abstract structures in English, Spanish, German and French. We developed cutting-edge neural network dependency parsers, a new kind of deep-syntactic transducers, and a fast rule-based frame-semantics parser that, unlike the state-of-the-art off-the-shelf counterpart, could be integrated in MULTISENSOR's Content Extraction Pipeline. We report on the evaluations carried out for the multilingual surface and deep parsers, and for the frame identification in English; all the objectives established in D1.2 have been attained. We also made some experiments on the methodology for frame identification through the use of existing multilingual lexical resources, which may open the way for large-scale multilingual frame assignment in a near future.

With respect to the **multimedia concept and event detection** task, in this deliverable we have presented a framework for the detection of predefined concepts/events specifically in video files. Due to the demanding nature of the problem under study, in this framework we have introduced the use of deep convolutional neural networks (DCNNs), in order to extract more sophisticated visual representations, compared to the local descriptors utilized in the previous versions of the framework. As demonstrated by the experiments that were conducted and presented in this deliverable, the DCNN features are much more suitable for this task, as for all employed concepts/events, the DCNN-based classification models significantly outperform the corresponding models that were trained on SIFT and RootSIFT local features. Finally, an additional advantage of the framework presented in this deliverable is the fact that it can be easily adapted to other domains through the use of transfer learning, by fine-tuning the DCNN parameters for new target datasets.

6 REFERENCES

Ahmad, K., L. Gillam and L. Tostevin. 1999. University of surrey participation in TREC8: Weirdness indexing for logical document extrapolation and retrieval (WILDER). In TREC Proceedings.

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

Baker, CF., Ch. J. Fillmore and J.B. Lowe. 1998. The Berkeley FrameNet project. In Proceedings of the 17th international conference on Computational linguistics-Volume 1, Association for Computational Linguistics, 86–90.

Ballesteros, M., B. Bohnet, S. Mille, and L. Wanner. 2014. Deep-Syntactic Parsing. In proceedings of COLING, 1402-1413.

Ballesteros, M., B. Bohnet, S. Mille, and L. Wanner. 2015. Data-driven deep-syntactic dependency parsing. Natural Language Engineering, 1–36.

Ballesteros, M., C. Dyer and N.Smith. 2015. Improved Transition-Based Parsing by Modeling Characters instead of Words with LSTMs . In proceedings of EMNLP, Lisbon, Portugal.

Ballesteros, M., C.Dyer, Y. Goldberg and N.Smith. 2016. Greedy Transition-based Dependency Parsing with Stack-LSTMs . Computational Linguistics. MIT Press.

Bay, H., A. Ess, T. Tuytelaars and L.Van Gool. 2008. “Speeded-Up Robust Features (SURF)”, Comput. Vis. Image Underst., vol. 110(3), pp. 346–359.

Björkelund, A., B. Bohnet, L. Hafdell and P. Nugues. 2010. A high-performance syntactic and semantic dependency parser. In Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 33-36.

Bohnet, B., J. Nivre, I. Boguslavsky, R.Farkas, F. Ginter and J.Hajic. 2013. Joint Morphological and Syntactic Analysis for Richly Inflected Languages,Transactions of the Association for Computational Linguistics.

Bosma, W. and Vossen, P. 2010. Bootstrapping Language Neutral Term Extraction, Proceedings of LREC 2010.

Candito, Marie, Benoît Crabbé, and Pascal Denis. 2010. "Statistical French dependency parsing: treebank conversion and first results." In Seventh International Conference on Language Resources and Evaluation-LREC 2010, pp. 1840-1847. European Language Resources Association (ELRA).

Corcoglioniti, F., M.Rospocher and A.P. Aprosio. 2016. A 2-phase Frame-based Knowledge Extraction Framework, 354–361.

Cucu H., Besacier L., Burileanu C. and Buzo A. 2012. ASR Domain Adaptation Methods for low-resourced Languages: Applications to Romanian Language. 20th European Signal Processing Conference (EUSIPCO).

Dyer, C., M.Ballesteros, W.Ling, A. Matthews and N. Smith. 2015. Transition-Based Dependency Parsing with Stack Long Short-Term Memory . In Proceedings of ACL (ACL 2015). Beijing, China.

Fedorenko D., N. Astrakhantsev and D.Turdakov. 2013. Automatic recognition of domain-specific terms: an experimental evaluation. In SYRCoDIS, pp. 15–23.

Frantzi, K.T., S. Ananiadou, and J. Tsujii. 1998. The c-value/nc-value method of automatic recognition for multi-word terms. In Research and advanced technology for digital libraries, pp. 585–604.

Girshick, R., J. Donahue, T. Darrell and J. Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587.

Hajič, Jan, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers et al. 2009. "The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages." In Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task, pp. 1-18. Association for Computational Linguistics.

Heigold G., Vanhoucke V., Senior A. and Nguyen P. 2013. Multilingual acoustic models using distributed deep neural networks. IEEE International Conference on Acoustics, Speech and Signal Processing.

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

Jia, Y., E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, pp. 675-678.

Kingsbury, P. and M. Palmer. 2002. From Tree-Bank to PropBank. In Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC), 1989-1993.

Krizhevsky, A., I. Sutskever and G.E.Hinton. 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097-1105.

Li J., Deng L., Haeb-Umbach R. and Gong, Y. 2016. Robust Automatic Speech Recognition. Academic Press, Elsevier.

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

Markatopoulou, F., V.Mezaris and I.Patras. 2015. “Cascade of classifiers based on binary, non-binary and deep convolutional network descriptors for video concept detection”. In Image Processing (ICIP), 2015 IEEE International Conference, pp. 1786-1790.

Markatopoulou, F., V.Mezaris and I.Patras. 2016. Online Multi-Task Learning for Semantic Concept Detection in Video. In Proceedings of IEEE Int. Conf. on Image Processing (ICIP 2016), Phoenix, AZ, USA.

Melčuk, I. 1988. Dependency Syntax: Theory and Practice. State University of New York Press, Albany.

Mille, S., A. Burga, L. Wanner. 2013. AnCora-UPF: A multi-level annotation of Spanish. In Proceedings of the 2nd International Conference on Dependency Linguistics (DepLing), Prague, Czech Republic, 217–226.

Mille, S, and L. Wanner. 2015. Towards large-coverage detailed lexical resources for data-to-text generation. In Proceedings of the First International Workshop on Data-to-text Generation.

Moro, A., A.Raganato and R.Navigli. 2014. Entity Linking meets Word Sense Disambiguation: A Unified Approach. Transactions Of The Association For Computational Linguistics, 2, 231-244.

Navigli, R., SP. Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence Journal, volume 193, 217–250.

Nivre, J., J. Hall, J. Nilsson, A. Chanev, G.Eryiğit, S.Kübler, S.Marinov and Marsi, E. 2007. MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering, 13(2):95-135.

Palmer, M. 2009. SemLink: Linking Propbank, VerbNet and FrameNet. In Proceedings of the Generative Lexicon Conference (GenLex-09), Pisa, Italy.

Presutti, V., F. Draicchio and A. Gangemi. 2012. "Knowledge extraction based on discourse representation theory and linguistic frames." In International Conference on Knowledge Engineering and Knowledge Management, pp. 114-129.

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

Russakovsky, O., J.Deng, H.Su, J.Krause, S. Satheesh, S.Ma, Z. Huang, A. Karpathy, A. Khosla, M.Bernstein and A.C. Berg. 2015. “Imagenet large scale visual recognition challenge”, International Journal of Computer Vision, 115(3), pp.211-252.

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

Simonyan, K. and A. Zisserman. 2014. “Very deep convolutional networks for large-scale image recognition”, arXiv preprint arXiv: 1409.1556.

Snoek, C.G., S. Cappallo, D. Fontijne, D. Julian, D.C. Koelma, P. Mettes, K.E.A. Sande, A. Sarah, H. Stokman, and R.B. Towal. 2015. “Qualcomm Research and University of Amsterdam at TRECVID 2015: Recognizing Concepts, Objects, and Events in Video”.

Srinivas, S., R.K. Sarvadevabhatla, K.R. Mopuri, N. Prabhu, S.S. Kruthiventi and R.V. Babu. 2016. “A Taxonomy of Deep Convolutional Neural Nets for Computer Vision”, arXiv preprint arXiv:1601.06615.

Szegedy, C., W.Liu, Y. Jia, P.Sermanet, S.Reed, D.Anguelov, D. Erhan, V.Vanhoucke and A.Rabinovich. 2015. “Going deeper with convolutions”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9.

Vedaldi, A. and K. Lenc. 2015, October. “Matconvnet: Convolutional neural networks for matlab”, In Proceedings of the 23rd ACM international conference on Multimedia, pp. 689-692, ACM.

Wiesler S., Richard A., Golik P., Schlüter, R. and Ney, H. 2014. The RWTH neural network toolkit for speech recognition. IEEE International Conference in Acoustics, Speech and Signal Processing.

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

Yosinski, J., J. Clune, Y. Bengio and H. Lipson. 2014. “How transferable are features in deep neural networks?”. In Advances in neural information processing systems, pp. 3320-3328.

Zhang, Z., J. Iria, C. Brewster, and F. Ciravegna. 2008. A comparative evaluation of term recognition algorithms. In LREC Proceedings.