MULTISENSOR

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

FP7-610411

D6.2

Content selection and presentation algorithms

Dissemination level:	Public
Contractual date of delivery:	Month 24, 31/10/2015
Actual date of delivery:	Month 25, 16/11/2015
Workpackage:	WP6 Summarisation and Content Delivery
Task:	T6.3 Content selection metrics T6.4 Content delivery procedures
Type:	Report
Approval Status:	Final Draft
Version:	1.0
Number of pages:	20
Filename:	D6.2_Summarisation_2015-11-16_v1.0.pdf

Abstract

This deliverable reports the development of methods for obtaining metrics or relevance for contents in the MULTISENSOR semantic repository, as part of task T6.3, and a first evaluation of the content delivery procedures developed within the scope of T6.4.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	1/10/2015	Layout proposal for comments	G. Casamayor (UPF)
0.2	26/10/2015	First draft	G. Casamayor (UPF)
0.3	10/11/2015	Draft for internal review	G. Casamayor, Simon Mille (UPF), Leo Wanner (UPF)
0.4	13/11/2015	Internal review	Boris Vaisman (LT)
1.0	16/11/2015	Final version	G. Casamayor (UPF)

Author list

Organisation	Name	Contact Information
UPF	Gerard Casamayor	gerard.casamayor@upf.edu
UPF	Simon Mille	simon.mille@upf.edu
UPF	Leo Wanner	leo.wanner@upf.edu

Executive Summary

This document presents work towards the design of metrics for the selection of contents from the MULTISENSOR semantic repository for the production of abstractive summaries, and an evaluation of the methods to improve the presentation of information in extractive summaries. Research on content selection metrics corresponds to task T6.3 of Work Package 6. We discuss a notion of empirical relevance for contents useful for the production of abstractive summaries and provide a formal description of a set of empirical metrics based on this notion of relevance. This is followed by a description of how these metrics will be applied to the conceptual summarisation of texts, as part of task T6.5.

This deliverable also describes current progress in task T6.4. The extractive summarisers presented in D6.1 have been extended with new annotations produced by the MULTISENSOR content extraction pipeline developed as part of Work Package 2. Several versions of the extended summariser are evaluated against the original version, taken here as a baseline, and the results are presented and discussed. This document closes with a review of the progress in both tasks T6.3 and T6.4, in relation to the topics covered in the Description of Work document.

Abbreviations and Acronyms

AS	Automatic Summarisation
DoW	Description of Work
GATE	General Architecture for Text Engineering
IDF	Inverse Document Frequency
IE	Information Extraction
JSON	JavaScript Object Notation
JSON-LD	JSON for Linking Data
LD	Linked Data
LOD	Linked Open Data
MT	Machine Translation
NE	Named Entity
NER	Named Entities Recognition
NIF	NLP Interchange Format
NLG	Natural Language Generation
NLP	Natural Language Processing
OWL	Web Ontology Language
RDF	Resource Description Format
REST	Representational State Transfer
TF	Term Frequency
WP	Work Package

Table of Contents

1 INTRODUCTION

In this deliverable we report the work done for tasks T6.3 and T6.4 of Work Package 6 (WP6) during the months 12 to 24 of the MULTISENSOR project. Task T6.3 covers research of content selection metrics, where content selection refers to the process by which information extracted from texts, stored in an RDF semantic repository, is assessed for the production of an abstractive summary. While abstractive summarisation can be applied to the production of a summary from contents extracted from one or more documents, having a semantic repository with interrelated contents obtained from a large volume of documents and linked to other datasets presents us with an opportunity to explore the production of summaries for one or more user-specified entities in the repository. The resulting summaries should contain the most relevant facts about these entities, regardless of what documents these facts were extracted from. We believe this type of summary is more interesting to end users than document-oriented summaries, because it enables the system to consider a much larger volume of information.

Starting from such a large volume of contents also requires effective mechanisms for evaluating contents. In this document, we present the research carried out to design empirical metrics of content relevance based on the statistical analysis of the semantic repository. This analysis is made possible by the fact that the repository also acts as a Linked Data (LD) corpus by virtue of using RDF models like NIF to store the links from the data extracted to the text fragments this data was obtained from.

While the metrics in T6.3 are directly applicable to abstractive summarisation, a similar approach can be adopted for extractive summarisation by adopting as features used to compare text fragments the semantic annotations produced by an analysis pipeline. There are some crucial differences in the type of metrics required for our approach to extractive summarisation, when compared to the planning of abstractive summaries. We provide a discussion on the topic and present a set of features and associated metrics that can be used to incorporate semantic annotations into our extractive summarisation pipelines. Some of the metrics we propose are then evaluated in terms of the summaries produced by the system compared to a gold standard of human-authored summaries.

This document is divided into a section for task T6.3 (Section **Error! Reference source not found.**) and another for task T6.4 (Section **Error! Reference source not found.**). These two sections are followed by a closing section where we draw some conclusions about the work done so far (Section 4).

2 CONTENT SELECTION METRICS

Content selection refers to the process of determining what contents in a data source should be used to generate a text. In the Natural language generation (NLG) field, this task is seen as part of a larger planning step where, starting from the data available to the sytem and some communicative goals (e.g. a user request), the general structure of the text (or speech) is defined. In this text planning step, content selection is often seen as the first task to be addressed, followed by a structuring task where the selected contents are sorted and arranged into a coherent whole.

Abstractive summarisation also involves the generation of a text in natural language (i.e. the summary), this time from data obtained from textual sources. In an abstractive summarisation system the selection of contents can be seen as the task of summarising data extracted from texts. Unlike the planning of texts from arbitrary data that may not be derived from textual sources, in the generation of abstractive summaries it is possible to preserve the links between the data and the text fragments it was obtained from, and use these links to create a corpus **of texts paired with data**. Such a corpus can then be the empirical basis for obtaining metrics for the selection of data. The MULTISENSOR content extraction pipeline and semantic repository have been designed to preserve and store these links, turning the semantic repository into a LD corpus.

Subsection 2.1 describes the MULTISENSOR semantic repository as a corpus that can be used to elicit content selection metrics. In subsection 2.2 , we provide a formal description of a set of metrics that can be obtained from the repository. Finally, in 2.3 we discuss how these metrics will be used in our implementation of a text planning module.

2.1 Using the MULTISENSOR semantic repository as a corpus

As a result of work in WP2, a text analysis pipeline has been deployed as part of the CEP. This pipeline produces **semantic annotations** encoded as RDF triples using semantic web vocabularies such as NIF[1], which are then stored into a central repository. The annotations contain links to both the text fragments they annotate and the larger documents they belong to. Annotations are currently being produced that mark mentions of Named Entities (NEs), terminological concepts (henceforth concepts), coreference links, predicate-argument structures, semantic classes of predicates and semantic roles adopted by their arguments, sentiment polarity and contextual information about the document as a whole (see deliverable D2.3).

Figure 1 shows a sentence with some of the annotations generated by the MULTISENSOR CEP. The sentence words are annotated with deep dependency relations that go from linguistic predicates to their arguments. The labels on these relations indicate argument numbers (*I*, *II*, *III*, etc.) for internal arguments of the predicate. External arguments are labelled *ATTR*. Internal arguments are those that the linguistic predicate requires in order to have a complete meaning and without each the sentence would look incomplete. Words also have annotated semantic entities which may be NEs and concepts from a database like

[1] http://persistence.uni-leipzig.org/nlp2rdf/

BabelNet[2] (Navigli and Ponzetto, 2012) (boxes with prefix *bn*), or relation types taken from a database like FrameNet[3] (Fillmore et al., 2002) (boxes with prefix *fn*). Relations in this annotation are the annotations of frames and their argument entities, which can be NEs, concepts or other relations. In the example there are three relations with frames Statement, Topic and Competition, the first one with two arguments, the second one with one and the last one with two. Whenever the CEP fails to assign a frame or entity to a word or multiword expression, their lemmas are used instead. Assuming the word *announced* in the example had no frame associated to it, its lemma *announce* would used as the relation type.

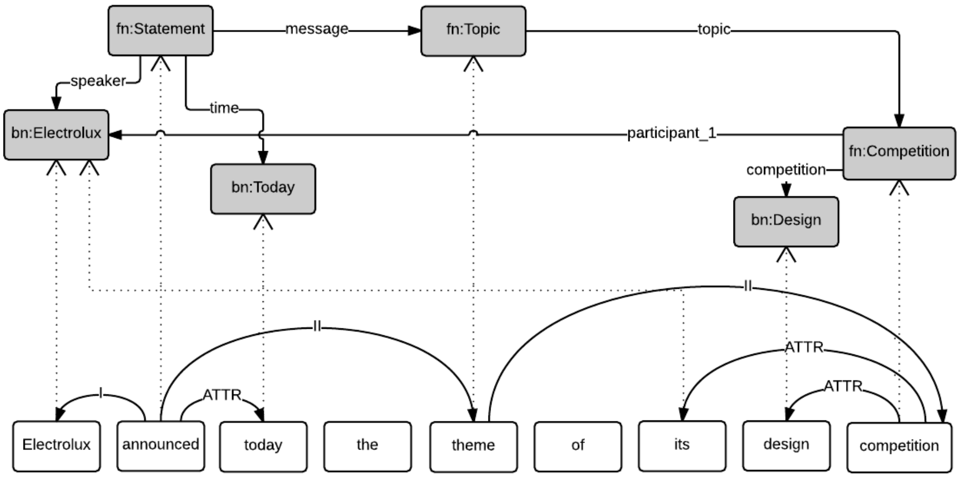

Figure 1: A sentence annotate with deep dependency relations, NEs, concepts and frames.

These annotations constitute the starting point for the generation of abstractive summaries. More precisely, annotations of predicates and their arguments are treated as the statements that constitute the basic unit for selection, structuring and linguistic generation. Thus, summarisation can be described as the process through which **relations** (derived from predicate-argument structures) are selected according to some criteria, sorted to maximise coherence and verbalised into a summary.

The choice of relations as the elementary units of summarisation responds to the requirements of linguistic generation. In contrast to single entities (NEs, concepts, isolated predicates), relations and their arguments can be more easily rendered as grammatical sentences or clauses. The fact that relations in the MULTISENSOR corpus always correspond to linguistic predicates certainly helps render them using natural language. Thus, given a relation and its participating entities, the linguistic predicates that realise it in the original texts can also be used to realise it again in the summary. Reusing the same predicates (or equivalent) as in the source texts also guarantees that the relations will provide arguments to fill each of the internal (required) arguments of the predicates, a requirement for grammatical and meaningful texts.

[2] http://babelnet.org/

[3] https://framenet.icsi.berkeley.edu/fndrupal/

When extracting relations from text, the CEP goes beyond identifying linguistic predicates and their arguments and tries to determine the situation the predicate denotes (the semantic type of relation) and the semantic roles adopted by its arguments in the situation denoted by the predicate. It does so by resorting to **FrameNet**, a lexical database of situational meanings indexed by linguistic predicates. Each entry in FrameNet, referred to as a frame, specifies a situation or relational meaning in terms of its participants, which are referred to as frame elements. The English version of FrameNet also comprises an index of lexical units (linguistic predicates) which are mapped to the frames denoted by them. Frames in FrameNet have been determined manually following a linguistic analysis of texts. Each frame generalises over similar meanings of linguistic predicates. Figure 1 shows three predicates annotated with frames and their arguments being labelled with frame elment roles.

The CEP uses NIF (Hellmann et al., 2013) and FrameNet RDF (Nuzzolese et al., 2011) to encode the semantic analysis of predicate-argument structures using RDF triples which are then stored into the system's semantic repository. While the relation extraction functionality uses various linguistic resources to attempt to determine (and disambiguate if necessary) what frame corresponds to a predicate-argument structure in a text, it ultimately relies on the FrameNet index of lexical units. Predicates not found in the index will therefore not be semantically analysed, which results in some relations produced by the system being devoid of semantic meaning. As illustrated in Figure 1 and explained above, the lemmatised predicate will be used instead as the relation type, and the arguments will be introduced with the deep syntactic labels (*I*, *II*, *III*, *ARG*).

Another relevant aspect of the relations found in the semantic repository is that in many cases their **arguments** are also semantically defined. By "defined" we mean that they have been linked to an entry in a dataset such as **BabelNet** (in the case of NEs and concepts) or to an entry in FrameNet (in the case relations). This is also shown in Figure 1, where the syntactic arguments in the deep dependency tree are annotated with BabelNet and FrameNet ids. Therefore, the actual semantic relation holds between these semantic annotations, whenever available. Notice that multiword expressions denoting NEs or concepts are already marked as a single token in the deep syntactic tree. In the CEP multiword expressions detected by the NER and concept extraction services are marked as a single token before running the syntactic parsing. This is done, in order to avoid unnecessary parsing of expressions that often have a non-compositional meaning and also to facilitate the extraction of relations pointing to the expressions as a whole rather than to individual words within them.

2.2 Metrics of relevance and ordering

The semantic annotations described in the previous subsection and stored in the semantic repository as a result of running the CEP on a volume of documents constitute a corpus of texts annotated with data. This corpus is currently being populated as part of WP2, and eventually will become large enough to act as a source of empirical knowledge about the relevance of contents for the generation of abstractive summaries. At the time of writing this deliverable the population of the semantic repository was in an early stage and no experiments or evaluation was possible for the metrics we are presenting here. The goal of the metrics is to inform an automated text planning module (described in the following subsection) about the relevance of contents, and also about partial ordering constraints that can be used to produce a sequence of content units with a minimum degree of internal coherence.

The notion of a priori relevance of contents (or text) is not very useful in an automatic summarisation setting, where relevance is always relative to a text or collection of texts to be summarised or, alternatively, one or more entries in a dataset for which a summary of most relevant facts must be generated. For these reason, we only consider **metrics for relative relevance**. Relevance and ordering metrics are useful for both extractive and abstractive summarisation. Indeed, a summary can be generated either from relevant contents using NLG methods, or by reusing the text fragments from which the contents being assessed were elicited from. The latter application of these metrics is described in section 3 covering task T6.4, albeit the metrics used there are always based on distance between content vectors assigned to sentences rather than a comparison between specific contents, as is the case here.

We propose the following metrics to be used for the selection of contents in the MULTISENSOR semantic repository:

1. Content co-occurrence $c(a_1, a_2)$ is the conditional probability of a pair of content items being annotated in the same document. The probability P is estimated with the relative frequencies of both annotations in the corpus obtained from counts $N(a_1, a_2)$:

$$c(a_1, a_2) = P(a_2|a_1) = \frac{P(a_1 \cap a_2)}{P(a_1)} \approx \frac{N(a_1, a_2)}{N(a_1)}$$

2. Weighted co-occurrence $c_w(a_1, a_2)$ is the interpolated conditional probability of a pair of content annotations being mentioned in the same document. The probability is estimated with the relative frequency of both annotations, each occurrence separated by a number of sentences $d \leq H$. Each count $N_d(a_1, a_2)$ used to estimate the probability is interpolated (weighted) by a factor dependent on the distance $\lambda_d = \frac{1}{d}$:

$$c_w(a_1, a_2) = P^{(H)}(a_2|a_1) = \frac{P^{(H)}(a_1 \cap a_2)}{HP(a_1)} \approx \frac{N^{(H)}(a_1, a_2)}{HN(a_1)} = \frac{\sum_{d=1}^{H} \lambda_d N_d(a_1, a_2)}{HN(a_1)}$$

3. Content ordering $\sigma(a_1, a_2)$ is the interpolated long-distance bigram (Bassiou and Kotropoulos, 2005) of a sequence of two content annotations. The probability is estimated with the relative frequency of annotations of a_2 following annotations of a_1, both annotations being separated by a number of sentences $d \leq H$. Each count $N_d\langle a_1, a_2\rangle$ of the sequence of two annotations used to estimate the probability is interpolated by a factor dependent on the distance between them $\lambda_d = \frac{1}{d}$:

$$\sigma(a_1, a_2) = P^{(H)}(a_2|a_1) = \frac{P^{(H)}(a_1 \cap a_2)}{HP(a_1)} \approx \frac{N^{(H)}\langle a_1, a_2\rangle}{HN(a_1)}$$
$$= \frac{\sum_{d=1}^{H} \lambda_d N_d\langle a_1, a_2\rangle}{HN(a_1)}$$

These three metrics can be calculated for pairs of three types of annotated contents:

1. BabelNet entries e
2. Relation names r which are either a FrameNet frame label or the lemma of a linguistic predicate
3. Full relations $f(r, \langle a_0 .. a_n\rangle)$ where r is a relation name and $\langle a_0 .. a_n\rangle$ is the sequence of content annotations (e or r) filling the core frame elements of the frame.

In this way, $c(e_1, e_2)$ indicates the co-ocurrence probability of two BabelNet synsets, $\sigma(r_1, r_2)$ the probability of two relation names appearing in order, $c_w(e_1, f(r, \langle e_2, e_3\rangle))$ the weighted co-occurrence probability between a BabelNet synset e_1 and a fully-specified relation $f(r, \langle e_2, e_3\rangle)$, etc.

Bearing in mind that the basic unit of the text planning module is a **full relation**, it becomes clear that only metrics comparing full relations are directly applicable to the task. Due to foreseen data scarcity in the annotation of relations, we plan on approximating the relevance of full relations on the basis of their constituent parts, the relation name and its arguments, e.g.:

$$c_w(e_1, f(r, \langle e_2, e_3\rangle)) \approx \frac{c_w(e_1, r) + c_w(e_1, e_2) + \ c_w(e_1, e_3)}{3}$$

2.3 Applying the metrics to text planning

For the planning of texts from data in the semantic repository we follow a data-driven and bottom-up approach, which is suited to the diffuse communicative goals of the summarisation task. Rather than having explicit goals which can be represented symbolically and reasoned about in a top-down fashion (e.g. by decomposing the goals), summarisation systems often follow a metric of relevance applied opportunistically on whatever texts or contents are available for summarisation. In addition to following a data-driven approach, we seek to exploit the connected nature of data in the semantic repository by adopting a graph representation of the contents which can be searched from one or more focal points, this points corresponding to entities of interest to the user for which the syummary is addressed. This search-based strategy is preferred over other approaches that operate on the whole data graph because it enables us to select contents in order, on the basis of contents already selected for inclusion in the summary. By selecting contents in this way we enforce a degree of coherence in the resulting summary.

Our graph representation of the input data is a focused **content graph** where nodes are atomic content units and edges indicate entity-sharing relations between content units. Content units are the elementary units of text planning and correspond to the full relations $f(r, \langle a_0 .. a_n\rangle)$ extracted from the source texts. An edge is established between two nodes whenever their argument lists share at least one BabelNet entry. Given one or more user-specified BabelNet entities, our goal is to produce a sequence of content units that are relevant to the query. Albeit our approach is focused on the production of abstractive summaries, in principle it would be possible to apply it to extractive summarisation by reusing the annotated text fragments that correspond to each content unit instead of generating new text using NLG methods.

Our notion of what content **relevance** stands for is empirical and based on the metrics defined in the previous section. Furthermore, rather than considering the a priori relevance of content units, we focus on the relevance of contents relative to a set of entities in a query for which the end-user of the system wishes to have a summary produced. In other words, this summary should contain the most relevant facts involving these entities, where facts in our semantic repository correspond to relations and their arguments.

Given a set of user-provided entities, a content graph and a set of metrics that allow us to assess the relevance of content units, text planning can be seen as a ranking problem, where the system must rank the nodes in the graph. In order to set a boundary on the number of contents to consider, the content graph can be reduced to a subgraph containing the nodes of the original graph with at least one reference to the entities in the user query, plus additional nodes connected with to the initial set of nodes up to a fixed depth. We refer to this subgraph as the **query graph**. Our problem formulation has similarities with the ranking of web pages addressed using connectivity-based and query-dependent algorithms like HITS (Kleinberg, 1998) or the content and query-sensitive version of PageRank by Richardson and Domingos (2007), which perform link analysis on a query graph. The fundamental principle of such algorithms is that the relevance of individual pages is propagated by the algorithm to neighbouring nodes of the graph following a link analysis, a notion that translates well to our setting. Similar ideas have been already applied to ontology summarisation (Zhang et al., 2007) and content selection with user-assigned weights (Demir et al., 2010).

We propose to use the metrics defined in the previous section to produce a weighted graph. Thus, text planning for the production of abstractive summaries in MULTISENSOR will proceed as follows:

1. Construction of a query graph: After a user query is formulated, a query graph will be generated from the contents in the semantic repository.
2. Application of metrics: A weighted version of the query graph will be produced by applying the content selection metrics to each node.
3. Weight distribution: Since the metrics are not likely to produce positive relevance assessments for all nodes in the graph, a dynamic algorithm will be run to distribute the weights following a recursive link analysis.
4. Content selection: The resulting weighted graph will then be navigated from the node with the highest relevance score, new content units being added to a sequence which constitutes the output of the text planning module. Ordering metrics will be taken into account during this stage. The length of the resulting summary will be determined by the number of content nodes added to the text plan.

The design and implementation of the tasks listed above correspond to task T6.5 and will be described in detail in D6.3. For the application of metrics and weight distribution we are experimenting with a method that draws ideas from link analysis algorithms and reinforcement learning methods. In particular, we formulate the content selection process as a markov decision process where links between nodes correspond to transitions between states, and relevance metrics constitute a reward function. In our experiments we are applying a sampling algorithm (e.g. Random Surfer, Monte Carlo method) that visits a sequence of nodes, obtains reward from visiting some of these nodes in the form of relevance assessments and updates the weights of all visited nodes according to the reward received and the transitions (links) between them.

3 CONTENT DELIVERY PROCEDURES

As described in the MULTISENSOR DoW, task T6.4 focuses on the development of methods for coherent presentation of information by exploiting the semantic information extracted from texts. In contrast to task T6.5, which covers the production of abstractive summaries through NLG methods, task T6.4 explores how to apply semantic data to the extractive (text-to-text) summarisation framework set up as part of task T6.1 and already integrated into the First Prototype.

In T6.1, we deployed single and multiple document summarisation pipelines that used the GATE[4]-based SUMMA tools[5] (Saggion, 2008) for extractive summarisation. These pipelines are completely separated from the CEP being developed as part of WP2, a situation that means no linguistic or semantic analysis being reused between the two. In this deliverable we report our efforts to change this situation and have SUMMA summarisation pipelines exploit the annotations produced by the MULTISENSOR text analysis pipeline.

While current progress has enabled us to effectively import the annotations in the semantic repository into SUMMA and derive summarisation metrics from them, the performance of the analysis modules, and more specifically of the NER and concept extraction modules, prevents the production of effective metrics for summarisation (see D2.3). This is due to a lack of coverage which leads to data sparsity (i.e. most sentences in the analysed texts have no disambiguated annotations of NEs nor concepts), and a poor precision which leads to noise (i.e. many annotations are wrongly disambiguated or refer to irrelevant concepts). For this reason, we have conducted an evaluation of the text-to-text summarisation using a state-of-the-art third-party tool, Babelfy[6] (Moro et al., 2014), which covers the functionality and replaces both the MULTISENSOR NER and concept extraction services.

3.1 Semantic features for extractive summarisation

In section 2.2 we presented a set of empirical metrics for the selection of contents in the MULTISENSOR semantic repository. These metrics estimate the relative relevance of pairs of content anntoations, NEs, concepts, frames or full relations. In their current form these metrics are not directly applicable to SUMMA given that the latter does not compare annotations individually but compares text fragments (e.g. sentences) using vectors which contain numerical evaluations of each annotation of a certain type found in the text. For instance, a SUMMA pipeline compares each sentence in a text to the first sentence by creating vector representations of each sentence which contain, for each lemmatised word, their tf-idf values.

In order to satisfy SUMMA requirements, we use the following features for creating sentence vectors:

1. TF-IDF for lemmas of tokens l (except stop words)

[4] https://gate.ac.uk/

[5] http://www.taln.upf.edu/pages/summa.upf/

[6] http://babelfy.org/

2. TF-IDF for annotations of BabelNet sysnets e produced by Babelfy
3. TF-IDF for triples of the form $\langle p, ds\_role(p, arg), arg \rangle$
4. TF-IDF for triples of the form $\langle p, arg \rangle$
5. TF-IDF for triples of the form $\langle r_f, fe(r_f, e), e \rangle$
6. TF-IDF for triples of the form $\langle r_f, e \rangle$

where p is a lemmatised predicate, arg is a lematised syntactic argument of the predicate, $ds\_role(p, arg)$ is the deep-syntactic label of the argument, r_f is the FrameNet frame of a relation, $fe(r_f, e)$ is the frame element label of the entity e in the frame r_f, and e is (the id of) a BabelNet synset.

The first metric is intended to be used in a baseline summarisation pipeline. The second one replaces the token lemmas with babelnet synsets annotated by Babelfy. The third and fourth metrics are based on the output of the deep dependency parser deployed as part of the MULTISENSOR analysis pipeline. Given a predicate detected by the parser, using these metrics involves generating a separate feature for each one of its arguments. Features for the first of them encode the full information about the predicate-argument pair consisting of the lemma of the predicate, the deep parser label of the argument (*I*, *II*, *III*, *IV*, *ATTR*) and the lemma of the argument word. The second of the metrics, which is based on the output of the deep parser (metric 4) produces compact and more general features where the argument label, arguably the least informative component of the predicate-argument relation, is dropped. This should help fighting problems of data scarsity. The last two metrics (numbers 5 and 6) use the output of the relation extraction module, which is also deployed as part of the MULTISENSOR analysis pipeline. Given a FrameNet-labelled n-ary relation, these metrics lead to the production of a feature for each relation-argument pair. The first of the two (metric 5) foresees features consisiting of the relation frame label, the argument frame element label and the argument id (either a BabelNet synset or a FrameNet id). The last metric (number 6) drops the frame element label, in order to produce more general features.

3.2 Evaluation of the extractive summarisation pipeline

We have compiled two corpora of 762 documents and 261 pairs of documents and human-authored summaries. All documents and summaries are in English, belong to use case 1.2 (household appliances), come from the pressrelations feed (NewsRadar repository[7]) and have been checked to meet certain quality requirements (no duplicates, minimal boilerplate text, no HTML, normalised end-of-lines, etc.). The first corpus has been used to obtain IDF tables for various types of annotations, while the second one has been used directly for the evaluation.

Both corpora have been processed using a pipeline, illustrated in Figure 2, set up specifically for this evaluation and consisting of four modules: linguistic pre-processing with Mate tools[8], entity linking with Babelfy REST service, deep dependency parsing with Mate tools

[7] http://www.pressrelations.com/pressrelations/index.cfm/en/newsradar

[8] https://code.google.com/p/mate-tools/

(Björkelund et al., 2010), and FrameNet-based n-ary relation extraction with our own module (described in D2.3). The annotations produced for the first corpus have been used to extract features of the six types described in the previous section, and the features indexed in a Solr[9] database for the calculation of IDF tables for each metric type.

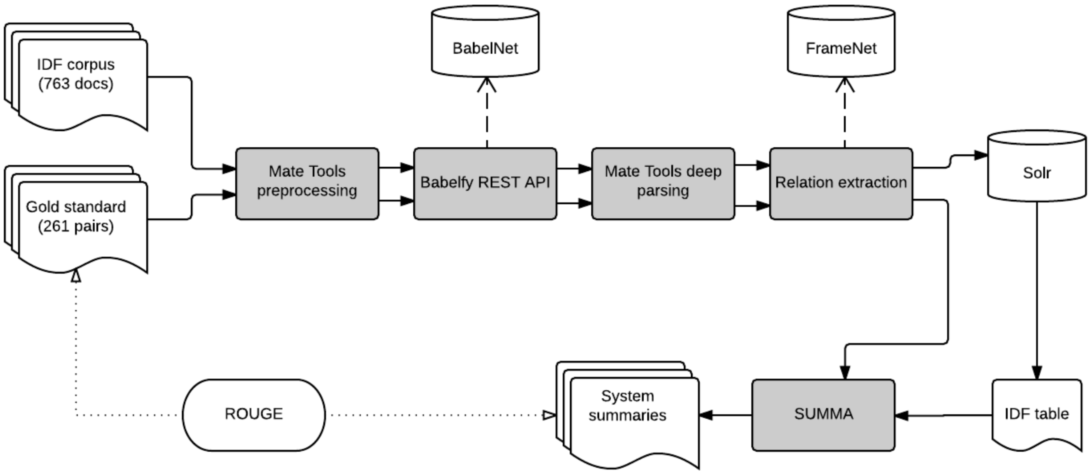

Figure 2: Pipeline for the evaluation of the summaries

The annotations in the orginal documents (not the summaries) of the second corpus have then been imported as GATE annotations and, together with the corresponding IDF table obtained from the larger corpus, used to run a SUMMA-based pipeline for single-document extractive summarisation. This pipeline uses a subset of the modules described in D6.1, namely:

1. NEs statistics: calculates and annotates basic statistics for each annotation:
    a. Number of times it appears in the document, sentence and paragraph.
    b. Inverted document frequency (idf), as measure of how much information a term provides, based on how common it is in a collection documents.
    c. Term frequency multiplied by inverted document frequency (tf*idf).
2. Term Frequency Filtering: filters out annotations that have a tf*idf below 10.
3. Vector Computation: creates a feature vector for the whole document containing pairs of annotations and tf*idf values.
4. Normalise Vector: normalises the document vector to [0..1].
5. Vector Computation: creates a feature vector for each sentence containing pairs of annotations and tf*idf values.
6. Normalise Vector: normalises the sentences vectors to [0..1].
7. Position scorer: adds a feature to each sentence indicating its relevance according to its position in the document.
8. Sentence document similarity: calculates and annotates for each sentence its similarity to the whole document, calculated by comparing their respective tf*idf vectors

[9] Solr is a fast, reliable and fault-taulerant open souce data base (http://lucene.apache.org/solr/)

9. Sentence term frequency scorer: sums and annotates the global frequencies in the document of each annotation in the sentence.
10. First sentence similarity: calculates and annotates for each sentence its similarity to the first sentence in the document, calculated by comparing their respective tf*idf vectors.
11. Simple summariser: performs a weighted combination of all sentence metrics calculated so far in order to obtain a relevance score for each sentence. The following metrics are used:
    a. Similarity to first sentence
    b. Similarity to whole document
    c. Sentence term frequency
    d. Position score

For this evaluation we have run the SUMMA pipeline three times using features and IDF tables for the metrics 1, 2, 3 and 6. In future evaluations we plan on evaluating all proposed metrics, and extend the evaluation to the other use cases. The evaluation has been conducted by comparing the summaries of the 261 texts generated by SUMMA to their respective human-authored summaries. The evaluation metric is unigram ROUGE (Chin-Yew, 2004) which compares lexical overlap of unigram vectors from sentences in the summary to be evaluated to the vectors of sentences in the gold standard summary. The results are shown in Table 1.

	Metric 1 l	**Metric 2** e	**Metric 3** $\langle p, ds\_role(p, arg), arg \rangle$	**Metric 6** $\langle r_f, e \rangle$
Unigram ROUGE	0.598	0.581	0.582	0.579

Table 1: ROUGE results of the evaluation of extractive summarisation using semantic features

The results do not provide any conclusive evidence supporting that replacing lemma-based features with semantic-based features improves substantially the performance of the extractive summarisation pipeline. There are various possible explanations for these results. First, poor performance of the semantic annotation tools may cause problems with data sparsity and noise, which limit the effectiveness of metrics based on semantic features. Further work on WP2 should result in better tools and higher-quality annotations. Using our concept extraction service may provide more relevant annotations than Babelfy, which is a general-purpose tool not tuned to the use case domains. The relation extraction module, on the other hand, is still under heavy development and has a poor disambiguation strategy which results in many incorrect assignments of frames to predicates with multiple senses.

Besides, due to time constraints we did not evaluate all metrics based on semantic annotations. It is possible that careful experimentation leads to features that are general enough to prevent data sparsity yet informative enough to effectively assess the relevance of sentences. Finally, there are other metrics and evaluation procedures for automatic summarisation systems beyond unigram ROUGE. In a future evaluation we intend to extend the metrics used in our evaluation and, if time and resources allow for it, perform a qualitative evaluation of the results.

4 CONCLUSIONS

This deliverable documents the work in WP6 for the months 12-24, focusing on tasks T6.3, content selection metrics and T6.4, content delivery procedures. We have provided a formal description of the metrics to be used for the text planning module of an abstractive summariser, and described the way these metrics will be applied for the selection and ordering of contents. Delays in the population of the knowledge base have prevented the experimental application of the metrics and subsequent empirical evaluation applied to text planning. Both will be conducted in the near future and reported in D6.3.

The extractive summarisation pipeline developed as part of task T6.1 has been extended in task T6.4 to include some of the information extracted by the MULTISENSOR content extraction pipeline. A first evaluation shows that the pipeline performs reasonable but fails to show if semantic features provide added value to the summarisation task. A more thorough evaluation is required using more advanced versions of the analysis tools developed in the scope of this project. Additionally, other types of extracted content may be taken into account that increase the quality and suitability of the generated summaries to user purposes, e.g. the results of sentiment analysis, contents extracted from multimedia contents.

A relevant outcome of the work carried out in this Work Package is how the results of the MULTISENSOR analysis pipeline and the various metrics that can be derived from them can be applied to both extractive and abstractive summarisation. User evaluations carried out in WP8 may shed more light into their real effect on the various summarisation pipelines.

Compared to the description of tasks T6.3 and T6.4 in the DoW, the work reported in this document fails to cover some specific points. Our metrics of relevance do not take into account user interest nor community opinion beyond taking into account a set of user-provided queries. Advancement in this area will depend on the performance of sentiment and context analysis in WP3. No information extracted from multimedia content is taking into account, which will require that the output of speech, image and video are stored into the semantic repository as part of WP2. As explained in D6.1, the lack of corpora of texts with human-authored summaries in languages other than English prevent the application of extractive summarisation methods to the rest of the MULTISENSOR languages. This is not the case for abstractive summarisation, where many of the analysis tools and the linguistic generation module of WP6 will support multiple languages, with MT being available as a fallback strategy. Finally, ordering and coherence in the generated summary have been discussed for tasks T6.3 and T6.5, but not for T6.4. Introducing coherence metrics requires substantial changes in the way SUMMA works and will be attempted in the later stages of this project.

5 REFERENCES

Bassiou, N. K., and Kotropoulos, C. L. 2005. “*Interpolated distanced bigram language models for robust word clustering*”, In Nonlinear Signal and Image Processing (NSIP 2005) IEEE-Eurasip, p. 12–12.

Björkelund, A., Bohnet, B., Hafdell, L., and Nugues, P. 2010. “*A high-performance syntactic and semantic dependency parser*”, In Coling 2010: Demonstration Volume, p. 33-36, Beijing.

Chin-Yew, L. 2004. "*Rouge: A package for automatic evaluation of summaries*”, In proceedings of the ACL-04 workshop Text summarisation branches out, Vol. 8.

Demir, S., Carberry, S., and McCoy, K. F. 2010. “*A discourse-aware graph-based content-selection framework*”, In Proceedings of the 6th International Natural Language Generation Conference, p. 17–25.

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

Hellmann, S., Lehmann, J., Auer, S., and Brümmer, M. 2013. “*Integrating NLP using linked data*”, Lecture Notes in Computer Science, p. 98–113.

Kleinberg, J. 1998. “*Authoritative Sources in a Hyperlinked Environment*”, In Proc. of the 9th ACM SIAM Symposium on Discrete Algorithms, p. 668–677.

Moro, A., Raganato, A., and Navigli, R. 2014. "*Entity linking meets word sense disambiguation: a unified approach*”, Transactions of the Association for Computational Linguistics 2, p. 231-244.

Navigli, R., and Ponzetto, S. P. 2012. “*BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network*”, Artificial Intelligence, 193, p. 217–250.

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, p. 41-48.

Richardson, M., and Domingos, P. 2007. “*The Intelligent surfer: Probabilistic Combination of Link and Content Information in PageRank*”, In NIPS, p. 1441-1448.

Saggion, H. 2008. “*A robust and adaptable summarisation tool*”, Traitement Automatique des Langues 49.2.

Zhang, X., Cheng, G., and Qu, Y. 2007. “*Ontology summarisation based on rdf sentence graph*”, In Proceedings of the 16th international conference on World Wide Web WWW 07, Vol. 2, p. 707.