Scalable Understanding of Multilingual MediA (SUMMA)

http://www.summa-project.eu

H2020 Research and Innovation Action
Number: 688139
D5.1 – Initial Progress Report on Natural Language Understanding

Nature	Report	**Work Package**	WP5
Due Date	31/07/2017	**Submission Date**	31/07/2017
Main authors	Afonso Mendes (Priberam), Pedro Balage (Priberam)		
Co-authors	Mariana Almeida (Priberam), Sebastião Miranda (Priberam), Nikos Papasarantopoulos (UEDIN), Shashi Narayan (UEDIN), Shay Cohen (UEDIN)		
Reviewers	Jeff Mitchell (UCL)		
Keywords	semantic parsing, AMR, summarization, sentiment analysis, monitoring		
Version Control			
v0.1	**Status**	Draft	20/07/2017
v0.2	**Status**	Draft	25/07/2017
v1.0	**Status**	Final	31/07/2017

Contents

List of Figures

Abstract

This document lays out the Initial Progress Report on Natural Language Understanding (WP5) for the first half of the *SUMMA* project. Overall, and for each task, we begin by stating the original intent (typically by quoting from the original proposal). We go on to describe what has actually been achieved.

Where possible technical descriptions are kept short, referring instead to reports or publications in annex.

1 Overview

This report consists of the initial scientific advances of the natural language understanding work package (WP5) in the *SUMMA* project. It also describes the first version of the *SUMMA* natural language understanding capabilities.

1.1 Objectives

(Text taken from the proposal) WP4 focused on the construction of a knowledge base from the stream of news articles that are fetched and pre-processed in WP3. This work package, WP5, complements WP3 and WP4 by refining the analysis of the news articles using "deep" natural language processing of the text. The term "deep" here refers to an analysis of language where the by-product is a graph structure, or a hierarchical structure that represents the meaning of the text in an abstract manner. As such, the machine learning techniques we use in this work package fall under the realm of structured prediction. From the application perspective, the focus of this work package is to organise news stories (which could potentially originate in several articles) into summaries that highlight the main events that thread the story together. These events will be marked with an aggregate score, denoting the sentiment towards them. The sentiment scores will be constructed based on a social media monitoring stream that the industrial partners, BBC and DW, will provide. From the technical perspective, the key engine for this work package is a storyline level semantics parsing algorithm, that will be subsequently used to synthesise stories into summaries. The two applications, the construction of a time line, and the indication of a sentiment score, will be based on that engine.

The main goals of WP5 are:

- the development of a fast, accurate and multilingual module for fine-grained semantic processing, capable of integrating several semantic formalisms (PropBank, FrameNet, AMR) into a unified graph representation at storyline level;
- the organisation of news stories, typically originated in a large number of news articles, into short summaries that highlight the main events that thread the story together;
- the matching of short events across different news articles and user-generated text (blogs and micro-blogs);
- the attachment of an aggregated sentiment score to each short event, taking into account all mentions to that event from user-generated text, with the possible use of variance information to detect events that are highly polarised.

1.2 Integration in the SUMMA pipeline

In order to achieve these goals, WP5 is divided into three main tasks which interact with the other components as illustrated in the Figure 1. Briefly, these tasks are the following:

Task T5.1: Storyline-Level Semantic Parsing: Extraction of a semantic graph representing an input storyline, which come from the clustering module (T3.4, WP3). This representation can subsequently aid other NLP tasks, such as summarization. This task is discussed in section 2, where we present our ongoing work on AMR and Propbank parsing, and storyline-level semantic parsing.

Task T5.2: Generation of Story Highlights: For each input storyline, generate a collection of story highlights which represent the most relevant content in the texts. This is accomplished via automatic summarization methods. This task is discussed in section 3, where we present our ongoing work on extractive and abstractive summarization with neural networks, CCA and ILP-based systems.

Task T5.3: Sentiment Analysis of a Story: Compute the aggregate sentiment score of social media sources (e.g., Tweeter) in respect to the generated story highlights. The final sentiment-annotated storyline highlight summaries are then displayed in the SUMMA platform interface. This task is discussed in section 4, where we discuss our current annotation efforts to create a dataset for sentiment analysis and summarization evaluation.

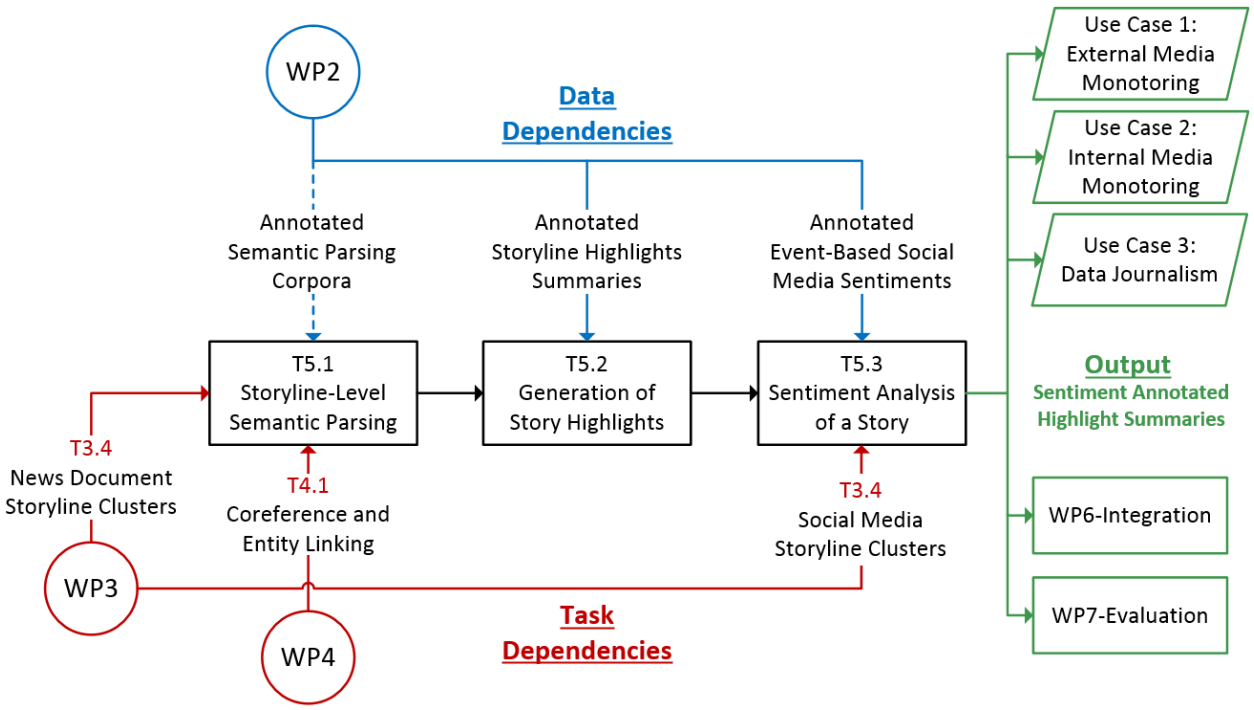

Figure 1: WP5 Tasks and their interaction with other *SUMMA* components

2 Storyline-level semantic parsing

2.1 Original description of work

Task T5.1: Storyline-Level Semantic Parsing (Partners: UEDIN, PRIB, LETA) (Text taken from the proposal) This task aims to develop statistical semantic parsers that go beyond sentences to operate at storyline level. The input to such a semantic parser is a collection of documents and articles belonging to a cluster, created in Task 3.4.

Such a parser takes the set of documents and creates a hierarchical structure that describes "who did what to whom" in the text. This parser, which is the key engine for this work package, requires some of the previous work packages to operate. For example, we could use the coreference recognisers and the entity linkers developed in Task 4.1 to collapse identical nodes in a graph representation. The output of this semantic parser will be a "collapsed" semantic graph that represents the entire story in an article or a set of articles, linking predicates to one or more arguments. This semantic graph contains n-ary relations that are deeper and more fine-grained than the ones produced in Task 4.2, since the goal is not knowledge base construction, but a fine-grained representation of a story line generated on the fly.

The output of this parser is a combinatorial structure. As such, its statistical inference algorithms and training algorithms require non-trivial combinatorial optimisation algorithms that could be computationally intensive. We will put a focus in this package on making the parser scalable, so that inference is more tractable, the statistical models learned are compact, and can be efficiently updated when more data is accessible. Both Priberam and the UEDIN have previous expertise in these topics (Martins et al., 2011b,a, 2013; Martins and Almeida, 2014; Cohen et al., 2013; Cohen and Collins, 2012). We will seek to integrate different semantic formalisms, such as PropBank (Kingsbury et al., 2002), FrameNet (Baker et al., 1998), and Abstract Meaning Representation (AMR; Banarescu et al. (2013)), into a universal scheme to facilitate comparison across different languages. This will build on past work done by our team in semantic role labelling (Das et al., 2014; Martins and Almeida, 2014).

For a restricted set of languages where annotated resources exist (English, Spanish, and German) supervised training will be used. For the remaining languages, we will use robust cross-lingual training techniques based on word-alignments (Ganchev and Das, 2013), or simply the English translations obtained via WP3. It is important to note that the semantic parser that is developed in this work package stands on its own merit, outside of the scope of media analytics. The NLP research community will greatly benefit from such a tool. As such, one of the deliverables will be a software package that provides the functionality of this parser.

2.2 Summary of progress

Semantic parsers enable the rich processing of text by extracting a semantic structure representing the meaning of natural language, typically a graph. In the context of the SUMMA project, the output of these parsers can be used as features to other NLP tasks, such as automatic text summarization (see section 3). These semantic parsing components are being developed for multiple languages and in different semantic formalisms, such as PropBank (Kingsbury et al., 2002) and Abstract Meaning Representation (AMR; Banarescu et al. (2013)).

The research progress in T5.1 can be divided in two main lines of work. On one side, LETA

and UEDIN developed state-of-the-art AMR parsers, which are a key technology to the project. That work is reported in the sections 2.4.2 and 2.4.1. On the other side, Priberam focused on the generation of semantic graphs at a storyline-level (as detailed in section 2.5) as well as improving Priberam semantic parser, TurboParser. As detailed next in section 2.3, some of these research efforts are already available as working modules in the *SUMMA* platform.

2.3 Semantic Parsing Capabilities in the *SUMMA* platform

Before we discuss the semantic parsing research efforts in further detail, we list the components which are already available SUMMA-wide:

AMREager: AMREager (Damonte et al., 2017) is an AMR parser being developed and maintained by the University of Edinburgh (UEDIN). The parser is an incremental left-to-right parser that works by scanning a sentence, and incrementally creating a graph that represents the meaning of the sentence. The parser was released on github,[1] and crosslingual extensions for it are provided (ongoing work).

RIGA AMR Parser: The CAMR+wrapper AMR parser (Barzdins and Gosko, 2016) has been developed by LETA and encapsulated into Docker container publicly accessible from the DockerHub:

docker run -it -v INPUTDIR:/data didzis/camrwrapper –input FILE –output FILE

The dockerised version of the CAMR+wrapper is among the fastest AMR parsers available, as it was specifically optimised for utilising multiple CPU cores and the slowest CAMR code segments were rewritten from scratch. AMR parsing speed is essential for the Big Data scaling of the SUMMA platform.

TurboParser: TurboParser (Das et al., 2014; Martins and Almeida, 2014) is a semantic parser maintained by Priberam, which received improvements in terms of computational speed and the ability to run in a multiprocessing environment, in order to scale for the SUMMA platform. The module has been dockerized and provided for SUMMA-wide usage in the BBC Box folder, while the TurboParser components are open source maintained in Priberam github repository [2]. Currently, the single-core performance is 2702 tokens/sec.

The description of these components from a software release perspective is further detailed in deliverable D5.2. In the following sections, we present the research efforts involved in the creation of some of these components, as well as other features or modules which are not yet released.

2.4 AMR Parsing Research

As listed in section 2.3, we pursued the research and development of two AMR parsing approaches, by UEDIN and LETA. In this section, we present an overview of these works, whereas a complete description is given in the referred publications.

[1] https://github.com/mdtux89/amr-eager
[2] Available at https://github.com/Priberam.

2.4.1 AMREager: An Incremental Parser for Abstract Meaning Representation

This subsection reports the results from the paper by Damonte et al. (2017), which presents an incremental parser for abstract meaning representation. A demo for this parser can be found here: http://cohort.inf.ed.ac.uk/amreager.html. Figure 2 gives an example of the result of running the parser on a simple sentence.

She described him as a curmudgeon

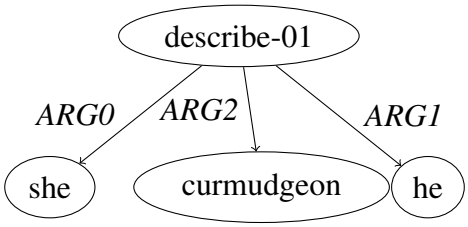

Figure 2: An example of a sentence parsed by AMREager ("she described him as a curmudgeon").

The parser works by scanning the input sentence left-to-right, word by word, and applying different transitions conditioned on the current state of the parser. The graph is generated incrementally while scanning the input that way.

In addition, we developed a set of metrics for fine-grained evaluation of AMR structures. The literature was lacking in that aspect – AMR parsers were evaluated by a single number (the Smatch score) that measures graph similarity. However, as AMR is composed of several different subtasks (such as named entity recognition and coreference resolution), we introduced a way to measure each of these subtasks separately. Indeed, we discovered that while one parser may be better than another on global Smatch score, it might perform worse on specific subtasks.

Table 1 compares the results of our parser to existing parsers: CAMR (Wang et al., 2015) and JAMR (Flanigan et al., 2014). As can be demonstrated there, while our parser still does not achieve the best results on the Smatch metric, it does best on other metrics.

Metric	JAMR ('14)	CAMR	JAMR ('16)	Ours
Smatch	58	63	**67**	64
Unlabeled	61	**69**	**69**	**69**
No WSD	58	64	**68**	65
NP-only	47	54	**58**	55
Reentrancy	38	41	**42**	41
Concepts	79	80	**83**	**83**
Named Ent.	75	75	79	**83**
Wikification	0	0	**75**	64
Negations	16	18	45	**48**
SRL	55	**60**	**60**	56

Table 1: Results of the different metrics for AMREager comparing to existing parsers. Best results in each row are in blue.

In addition, as part of the work on AMREager, we developed a method to do cross-lingual AMR parsing. This work capitalises on the idea that AMR structures are supposed to be a canonical

representation of a given sentence, and as such, they can be transferred across sentences in different languages, as long as those sentences are translations of each other.

We developed the cross-lingual parser for German, Spanish, Italian and Chinese. Smatch scores for these parsers are given in Figure 2. Our plan is to further extend this crosslingual parser so that it covers the languages in SUMMA. Fortunately, building such a crosslingual parser does not require much resources: it mostly requires an NLP process pipeline in the target language and a large parallel corpus between English and the target language.

	Silver	Gold
Italian	0.45	0.45
Spanish	0.44	0.44
German	0.45	0.43
Chinese	0.36	0.40

Table 2: Results of the cross-lingual parser.

2.4.2 RIGA AMR Parser: Impact of Smatch Extensions and Character-Level Neural Translation on AMR Parsing Accuracy

As described in detail in the paper "RIGA at SemEval-2016 Task 8: Impact of Smatch Extensions and Character-Level Neural Translation on AMR Parsing Accuracy" (Barzdins and Gosko, 2016), LETA has developed three novel AMR parsing extensions:

1. Visual SMATCH with C6.0 rule-based classifier reporting patterns for systematic AMR parser errors in the scored AMR graphs.
2. Per-sentence SMATCH combined with an ensemble method for selecting the best AMR graph among the set of AMR graphs.
3. AMR parsing by character-level neural translation, which attains a surprising 7% gain over the word-level neural translation.

The first extension resulted in 4% gain over the state-of-art CAMR baseline parser (Wang et al., 2015) through adding to it a manually crafted wrapper fixing the identified CAMR parser errors. The second and third extensions yielded further 0.4% gain when applied to outputs of three AMR parsers: two non deterministic instances of the CAMR+wrapper parser and one instance of a novel character-level neural translation based AMR parser.

These CAMR parser extensions were sufficient for us to win (along with the original CAMR team) the SemEval-2016 Task-8 on "Meaning Representation Parsing". We achieved Smatch F1=62% on the official SemEval-2016 Task-8 scoring set and F1=67% on the LDC2015E86 test set.

2.5 Storyline Semantic Graph Parsing

The main purpose of generating a storyline-level semantic graph is to assist the automatic summarization of document storylines (or clusters, as generated by T3.4, WP3). The idea is to gather all

the semantic information of a particular storyline in a unified semantic graph and afterwards pick the subgraph which would better summarize the storyline.

Our initial experiments, led by Priberam, consisted in linking nodes from the semantic graphs produced by sentence-level semantic parsers. We experimented with two different semantic formalisms: PropBank (Kingsbury et al., 2002) and Abstract Meaning Representation (AMR; Banarescu et al. (2013)).

For the PropBank semantic graphs, we used Priberam's version of TurboParser (Das et al., 2014; Martins and Almeida, 2014), in order to generate semantic triples in the form of *subject-verb-object*. These triples are built by extracting for each verb present in the text, the respective agent (Propbank argument 0) and patient (Propbank argument 1). The triples were inserted into the storyline graph and identical nodes were merged and their weight increased[3]. Figure 3 shows the semantic graph for a storyline with a zoom on the entity "A380", which is the central entity in the storyline about the Airbus company and its new plane.

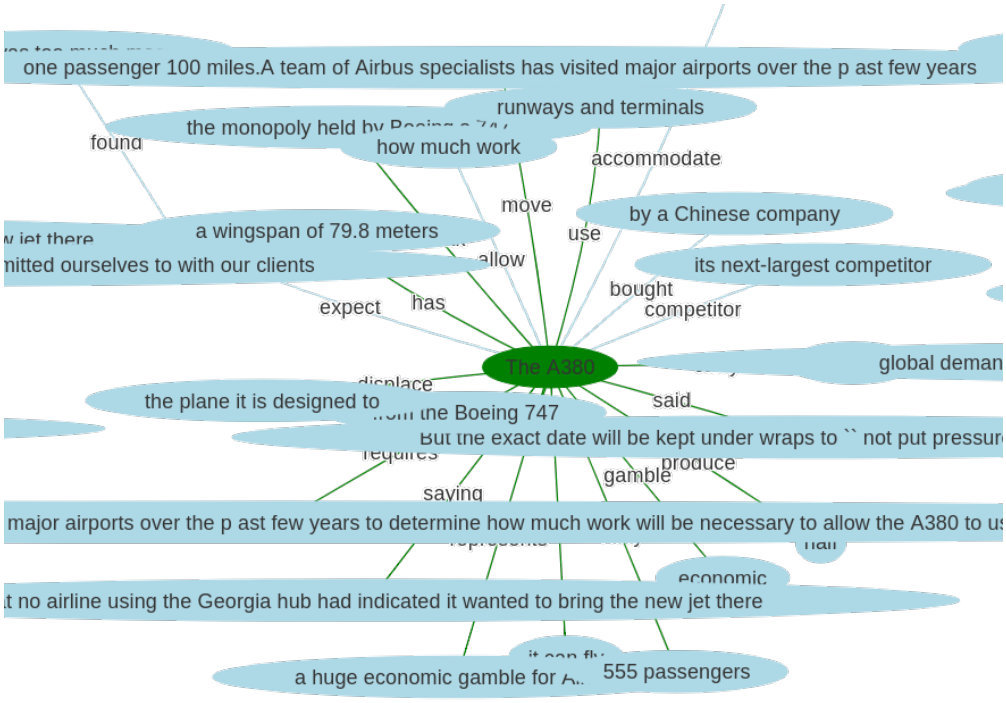

Figure 3: Entity "The A380" in the semantic graph

For testing with AMR, we used LETA's CAMR wrapper which provided a more fine grained semantic annotation compared with the PropBank triples. Figure 4 shows the storyline semantic graph built using AMR. The nodes are AMR nodes and the edges are the AMR functions between the nodes. In this graph, only nodes with nouns, verbs and entities are connected. The human summary that this graph covers is included in the figure for reference.

The objective of this component is to serve as input for the next component, generation of story highlights. Accordingly, evaluation of this component is performed extrinsically by evaluating the ROUGE score in highlight summaries produced by taking information from the semantic graph.

[3] The proposal mentions the use of entity linkers and coreference to connect the nodes. Since these tools were not yet ready to incorporate into this research, we adopted a simpler approach by connecting the nodes using text similarity.

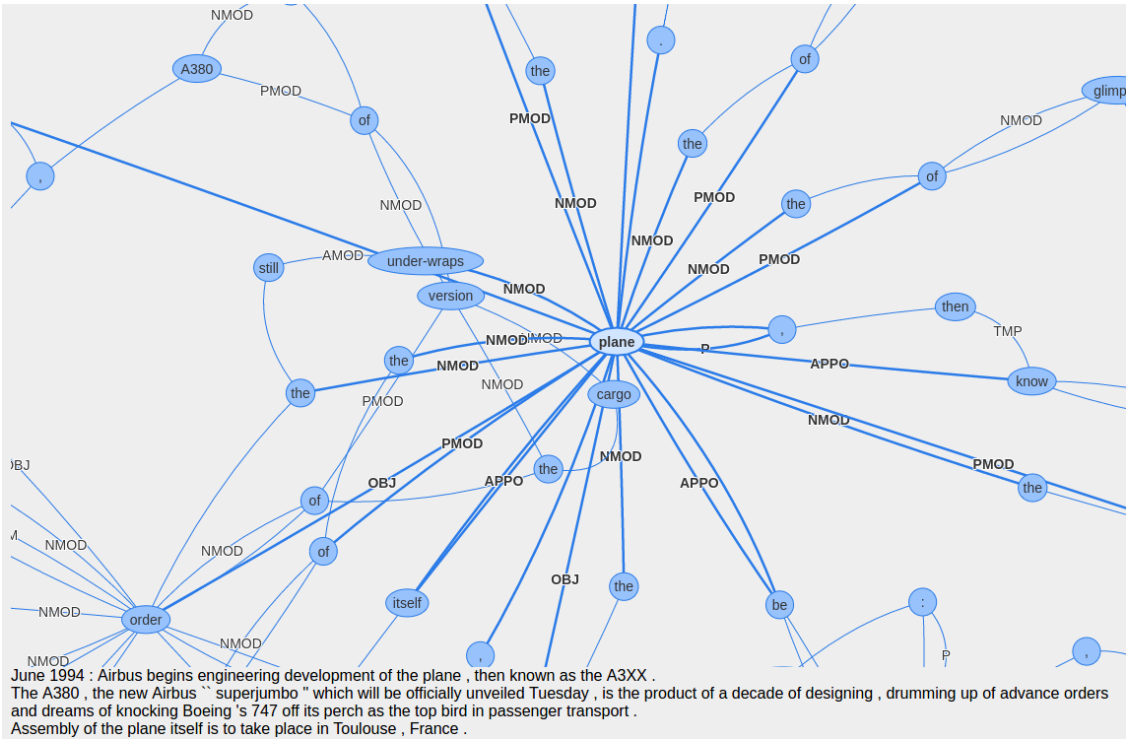

Figure 4: Semantic graph built with the AMR nodes and their relations.

The experiments reported in sections 3.3 and 3.4 show how this component may be incorporated into the highlight generation module.

2.6 Final Remarks and Future work

In the context of the SUMMA project, we pursued the research and development of two AMR parsers, UEDIN's AMREager and LETA's CAMR wrapper. Additionally, Priberam improved their existing semantic role labelling parser, TurboParser, to scale better in SUMMA. These components serve not only as standalone research outputs but also as supporting NLP components for the rest of the SUMMA pipeline. In particular, we experimented with storyline-level semantic graph construction in order to enable storyline summarization, as will be further discussed in the next section. Besides improving existing modules, future work includes further experimentation with the use of semantic graphs in the summarization task.

2.7 Publications

- Damonte, M., Cohen, S. B., and Satta, G. (2017). An Incremental Parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain, April 3-7, 2017. Association for Computational Linguistics

- Barzdins, G. and Gosko, D. (2016). RIGA at SemEval-2016 Task 8: Impact of Smatch Extensions and Character-Level Neural Translation on AMR Parsing Accuracy. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1143–1147

3 Generation of Story Highlights

3.1 Original description of work

Task T5.2: Generation of Story Highlights (Partners: PRIB, LETA, UEDIN) (Text taken from the proposal) The previous task constructs the key engine for this work package, which will be further used in downstream applications.

The first application is that of story time-line development. Here, the goal is to take the output of the semantic parser, and synthesise a coherent summary of all events that occur in the story according to the semantic parser. The key motivation behind such summarization is to enable a person monitoring the media see a global picture and evolution of a story.

To accomplish this, we will adapt coverage-based summarization models (Yih et al., 2007; Gillick et al., 2008) to cope with rich concept representations extracted from the semantic graphs obtained in Task 5.1.

A challenge is to identify redundant and synonymic events across news articles and assess their relevance – usually formulated as a combinatorial max-cover problem. We will build on prior work done by our team in scaling up compressive coverage-based summarisers (Almeida and Martins, 2013), which we will adapt to handle predicate argument structures. The key novelty of this task is to focus on events as the units that compose the summary, a step forward from previous extractive approaches. The final summary will contain the most salient events, where equivalent ones are collapsed. This collapsing will be guided by frame information or directly captured by the abstract meaning representations produced in Task 5.1. We will also use previous work by researchers at Edinburgh, who showed how symbolic representations can be combined with continuous ones to identify synonymic predicates (Lewis and Steedman, 2013).

A key component will be the serialisation of the events according to a timeline, so that the summary represents a progression in time. Here, we might make use of previous work developed by the consortium (Abend et al., 2015) for the goal of this serialisation.

3.2 Summary of progress

One of the main novelties of the SUMMA platform will be the ability to provide storyline highlight summaries across a linked set of related stories (storyline), which are the clusters produced by the clustering module (T3.4, WP3). The main goal of generating storyline highlight summaries is to enable efficient news monitoring and content discovery, and thus satisfy the SUMMA BBC and DW use-cases.

The summarization field has gained increased attention recently with the release of bigger datasets (Napoles et al., 2012; Hermann et al., 2015) and data-driven approaches based on neural networks (Rush et al., 2015; Cheng and Lapata, 2016; See et al., 2017; Paulus et al., 2017).

The development of this component in *SUMMA* focused on three main approaches: basic extractive coverage models, semantic graph modelling for highlight generation; and deep learning techniques for automatic highlight generation. This last one has been given more focus since it is where the lastest advancements in the field arose and so promises the potential for better results. We intend to make these various approaches available as pluggable modules in the SUMMA pipeline, giving the user the possibility to choose between the different advantages of each approach.

Section 3.3 reports the usage of Priberam's coverage-based extractive models for the task, showing that semantic knowledge extracted from the semantic graph may improve the actual methods. Subsequently, section 3.4 shows a direct graph summarization approach attempted by Priberam. Then, in section 3.5, we report LETA's work with AMR-to-text generation approach which contributes to the problem of generating the highlights from a semantic graph. In section 3.6.1, we approach the problem of extractive summarization with neural networks and side information. Finally, Section 3.6.2 shows the generation of abstractive highlights guided by extractive algorithms.

Evaluation and State-of-art

The standard metric to evaluate summarization is ROUGE (Lin, 2004), an n-gram overlap of the predicted summary against the reference human summaries. Other common metrics are Linguistic Assessment and Pyramid (Nenkova et al., 2007) which require human efforts to label the system's output summary.

For conducting the evaluation we considered datasets provided by the main shared tasks on summarization (TAC and DUC) as well as new datasets that explore the concept of highlights summary such as the CNN/Daily Mail dataset (Hermann et al., 2015; Cheng and Lapata, 2016). The former has the advantage of exploring multi-document clusters while the latter provides highlights summaries.

In this project context, the ideal dataset would cover both multi-documents and highlights summaries. In the absence of such dataset, which would evaluate the project main goals, we created our own dataset which will be used as a test set for evaluating this component. The dataset creation is covered in subsection 4.3.

The description of these components from a software release perspective is further detailed in deliverable D5.2. In the following sections, we present the research efforts involved in the creation of some of these components, as well as other features or modules which are not yet released.

3.3 Coverage-based extractive summarization

In order to summarise a storyline into a list of highlights we took inspiration in prior work in multi-document summarization using coverage-based summarization models (Yih et al., 2007; Berg-Kirkpatrick et al., 2011). These models extract a set of relevant concepts (e.g. n-gram keywords) from the original documents, assign them a salience score, and then seek the summary that maximises the overall score. This can be framed as a global combinatorial optimisation problem.

Formally, in extractive summarization, a set of sentences $\mathcal{D} := \{s_1, \ldots, s_N\}$ belonging to one or more documents is extracted into a subset $\mathcal{S} \subseteq \mathcal{D}$ that conveys a good summary of $\mathcal{D}$ and whose total number of words does not exceed a size restriction B.

With the vector $\boldsymbol{y} := \langle y_n \rangle_{n=1}^{N}$ representing the extractive summary, $y_n = 1$ leads to a selected sentence while $y_n = 0$ do not. Let L_n be the number of words of the nth sentence. By designing a quality score function $g : \{0,1\}^N \to \mathbb{R}$, this lead us to a knapsack formulation of the problem:

$$\begin{aligned} \text{maximize} \quad & g(\boldsymbol{y}) \\ \text{w.r.t.} \quad & \boldsymbol{y} \in \{0,1\}^N \\ \text{s.t.} \quad & \textstyle\sum_{n=1}^{N} L_n y_n \leq B. \end{aligned} \tag{1}$$

where L_n is the number of words of the nth sentence.

Essentially, two models have been proposed for designing $g(\boldsymbol{y})$, both capturing the idea that a good summary is one that selects sentences that individually contribute with "relevant" information, while collectively having small "redundancy": maximal marginal relevance (Carbonell and Goldstein, 1998; McDonald, 2007) and coverage-based (Filatova and Hatzivassiloglou, 2004; Yih et al., 2007; Gillick et al., 2008; Almeida and Martins, 2013). The latter attempts to maximise the information coverage by introducing some notion of "concepts" and then seeking a set of sentences that covers as many concepts as possible; redundancy is automatically penalised since redundant sentences, as a whole, cover fewer concepts.

Coverage-based extractive summarization can be formalised as follows.

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{m=1}^{M} \sigma_m u_m \\ \text{w.r.t.} \quad & \boldsymbol{y} \in \{0,1\}^N, \ \boldsymbol{u} \in \{0,1\}^M \\ \text{s.t.} \quad & u_m = \textstyle\bigvee_{n \in \mathcal{I}_m} y_n, \ \forall m \in [M] \\ & \textstyle\sum_{n=1}^{N} L_n y_n \leq B, \end{aligned} \tag{2}$$

where σ_m be a relevance score assigned to the mth concept, $\mathcal{I}_m \subseteq \{1, \ldots, N\}$ keep the indices of the sentences in which this concept occurs and $u_m(\boldsymbol{y}) := \bigvee_{n \in \mathcal{I}_m} y_n$ is a Boolean function that indicates whether the mth concept is present in the summary. This formalization above can be converted into an integer linear program (ILP) and addressed with off-the-shelf solvers (Gillick et al., 2008; Martins et al., 2015).

In our architecture, we implemented the coverage-based extractive summarization system expressed by Equation 2, using bigrams as the concept unit (Almeida and Martins, 2013). We conducted three experiments with the coverage formulation. The first experiment uses a basic coverage model based on the formulation in Equation 2. The second experiment complements the previous experiment by adding frequency, positional and categorical features for weighting each concept. The weights are learnt in an online SVM algorithm based on a structured prediction approach. The third experiment included new features for the concepts extracted from the storyline semantic graph reported in subsection 2.5. For each concept that exists as a node in the semantic graph, we extracted the graph metrics of node weight, centrality and PageRank. Table 3 shows the ROUGE recall scores for these experiments in the TAC 2008 corpus.

summariser	R1	R2	RL
Cov-based Extractive	36.43	9.9	32.4
Cov-based Extractive + Feat.	37.0	10.6	33.0
Cov-based Extractive + Sem.Feat	**37.1**	**10.7**	**33.1**

Table 3: ROUGE score (recall) for multi-document summarization with 100 words in TAC 2008 corpus.

Although these results provide an initial demonstration of the utility of semantic knowledge in automatic summarization, the semantic network plays only the marginal role of providing features to the main coverage maximization based approach. Conversely, what we seek in the project is to confer a more determinant role to the semantic network, which will be explored using graph-based

summarization in the next section. In addition, we also seek to have a system which generates summaries in the form of story highlights, which is the desired format for the project.

To face these challenges, new methods should be introduced in order to leverage the potential of the semantic graph and to generate highlights. The next section shows initial experiments with a direct summarization from the semantic graph.

3.4 Graph-based summarization

Following on from the previous section, we implemented a coverage-based method that selects semantic triples from the semantic network, rather than extracting summary sentences from the source. Our goal is to adapt the algorithm to put the semantic network at its core, thus utilising the full potential of this semantic information.

As before, we approached the problem by defining the vector $\boldsymbol{y} := \langle y_n \rangle_{n=1}^{N}$ representing the possible triples extracted from the semantic graph in the form of predicate argument structures, where $y_n = 1$ leads to a selected triple while $y_n = 0$ does not. Then, L_n is the number of the words with the surface realisation for the triple extracted. We kept the base problem formulation (Equation 2) unchanged. In this formulation, we now consider the concepts not only as simple bigrams but also as nodes from the semantic graph.

Since our target is now to produce story highlights, we conducted experiments with the CNN/Daily Mail corpus previously collected by Cheng and Lapata (2016). An example of summary generated by this method is illustrated in Figure 5.

This method was evaluated using the ROUGE metric in the CNN test-set corpus. Accordingly, this method achieved ROUGE F1 scores of R1=21.21 and R2=5.46, whereas a first-sentence baseline had ROUGE scores of R1=23.89 and R2=7.75. Although according to the ROUGE metric this current approach does not beat the baseline, it is important to note that this research direction is still in an initial phase and that more experiments should be conducted in order to better understand the possibilities of this approach. One of the main challenges we face is the surface realisation (text generation) from the given triples. In the example, we just joined the agent, verb and patient arguments leading to many cohesion problems, and low ROUGE scores. In the project we plan to explore other methods for graph-to-text generation, such as the one reported in the next section.

Even though the graph-based summary presents some errors, the summary format is closer to the desired highlights, which is the main reason to experiment with this approach. One possible future direction is to adapt this approach to work with deep learning methods, which would lead to better results due their capacity to more effectively model the highlight generation task.

In the next section, we describe the work we conducted on AMR-to-text generation which should be very beneficial for this line of work.

3.5 AMR-to-text generation approach

In parallel with the semantic graph summarization, an important feature is the ability of the system to automatically produce a good quality summary. The previous work of Greenbacker (2011) showed an initial approach to merge semantic graphs in order to build an abstractive summariser. However, their work lacked a means of converting AMR graphs into textual representations.

Following the similar task in text-to-AMR parsing (May, 2016), a recent shared task at SemEval

COV-BASED EXTRACTIVE SUMMARY :

CNN) – Singapore Airlines received its first A380 superjumbo on 15 October Monday , ready for the first commercial flight between Singapore and Sydney on Thursday 25 October .
Two years later in January 1993 , Boeing joined the Airbus partners Aerospatiale , Daimler-Benz Aerospace Airbus , Construcciones Aeronauticas and British Aerospace in the studies .
On April 27 2005 the A380 took its maiden flight from Toulouse to Bay of Biscay for four hours .
In October 2006 , Christian Streiff resigned as Airbus CEO and Louis Gallois took over .
But Fedex cancelled its order of 10 A380s on November 8 2006 .

GRAPH-BASED SUMMARIZATION:

Airbus began talks with major international carriers about requirements for a super-large passenger aircraft
Airbus talks about requirements for a super-large passenger aircraft
Airbus said customers would get deliveries up to two years late
Airbus said it was six months behind delivery schedule
Airbus sale 10,000 jobs cuts and plants
Airbus formed its Large Aircraft Division
Airbus received its first fresh order
Airbus announced delays for third time
its step to certification on November 13 2006 with the technical route proving mission
its stake in Airbus
its order of 10 A380s
Singapore Airlines received on 15
Singapore Airlines bought nine more A380s

Figure 5: Example of summary produced up to 100 words.

2017 (Task 9, Subtask 2) unveils the state-of-the-art in AMR-to-text generation (May and Priyadarshi, 2017).

A common general approach of most systems is to convert the given AMR graph into a tree, and then to generate a surface realisation from the tree. For instance, Flanigan et al. (2016) convert AMR graphs into spanning trees, and decode the trees using a weighted tree-to-string transducer and an n-gram language model, while Lampouras and Vlachos (2017) transform AMR graphs into dependency trees, and linearize the trees using a classifier for ordering the nodes in each subtree.

In our approach (Gruzitis et al., 2017; Gruzitis and Barzdins, 2016), we use Grammatical Framework (GF) as the intermediate tree representation. GF (Ranta, 2011) is a grammar formalism and a technology for implementing computational multilingual grammars that are particularly well suited for language generation.

The key feature of GF is the division between abstract and concrete syntaxes. A concrete syntax defines the surface realisation of the abstract syntax for a particular language. The same abstract syntax can be equipped with many concrete syntaxes, making the grammar multilingual. Another key feature of GF is its wide-coverage resource grammar library (RGL) with a shared abstract syntax.

The idea is to transform AMR graphs (story highlights) to GF abstract syntax trees (AST), leaving the linearization of the acquired ASTs to the existing English resource grammar. The linearization

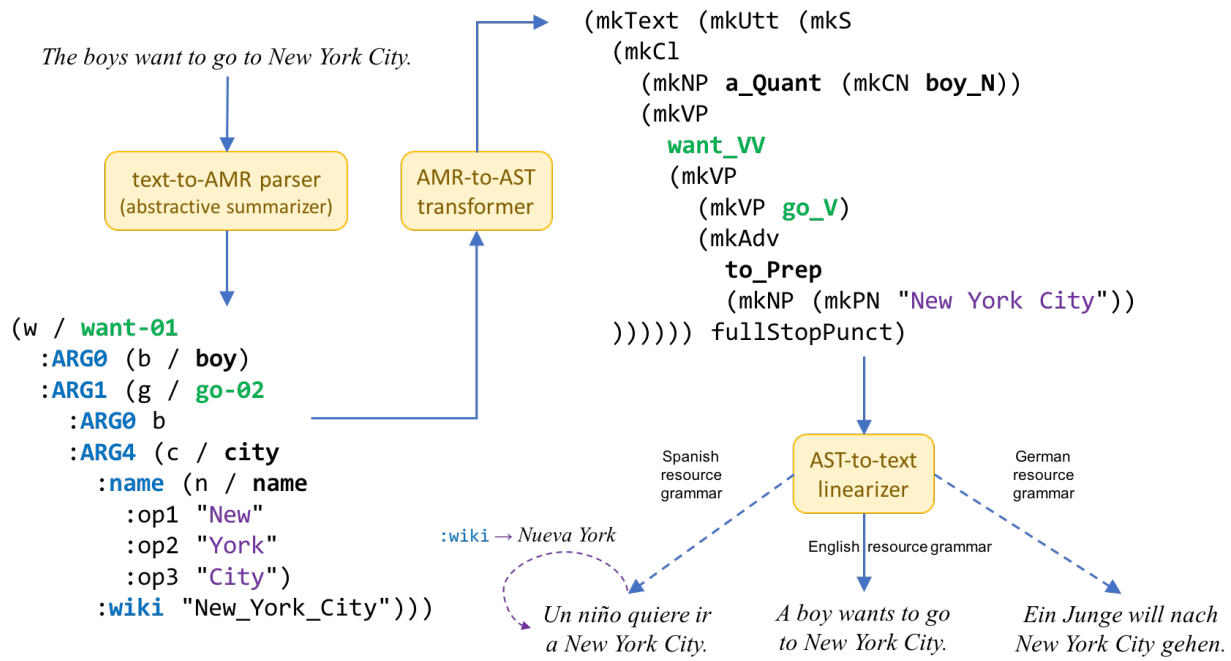

Figure 6: AMR to Grammatical Framework to Text

of an AST resolves the word order, word forms (syntactic agreement), function words, etc. Figure 6 outlines the architecture of our approach.

Since RGL supports many more languages (30+), this approach is relatively easily extensible to multilingual AMR-to-text generation, given a wide-coverage translation lexicon (currently available for 15+ languages). The alternative approaches focus solely on AMR-to-English generation.

The current bottleneck of our approach is the rule-based transformation from AMR graphs to AST trees. Because the coverage of our hand-crafted AMR-to-AST transformation rules is currently far from complete, we use the JAMR generator Flanigan et al. (2016) as a default option for AMRs that are not fully covered by the current rule set. Nevertheless, the SemEval 2017 Task 9 results show that the combined approach outperforms the JAMR generator, achieving the human Trueskill score of 1.03, compared to Trueskill 0.82 achieved by JAMR generator alone, which is otherwise the best performing AMR-to-text generation system.

3.6 Summarization with neural networks

As described earlier, we also started to work with neural-based approaches, since they have been gaining a lot of traction in the literature. In this section, we discuss two different approaches to the problem, one considering extractive summarization (selecting whole sentences), whereas another considers abstractive summarization (allows text generation).

3.6.1 Neural Extractive summarization with Side Information

In this approach, we propose to explore side information in the context of single-document extractive summarization. We develop a framework for single-document summarization composed of a

hierarchical document encoder and an attention-based extractor with attention over side information. We evaluate our models on a large scale news dataset. We show that extractive summarization with side information consistently outperforms alternatives that do not use side information, in terms of both informativeness and fluency. We report state-of-the-art results on the CNN dataset. We refer the reader to Narayan et al. (2017b) or a running demonstration of our system at http://cohort.inf.ed.ac.uk/sidenet.html for more details. Below, we summarize our motivation, model and results.

South Korean Prime Minister Lee Wan-koo offers to resign

Seoul (CNN) South Korea's Prime Minister Lee Wan-koo offered to resign on Monday amid a growing political scandal.

Lee will stay in his official role until South Korean President Park Geun-hye accepts his resignation. He has transferred his role of chairing Cabinet meetings to the deputy prime minister for the time being, according to his office.

Park heard about the resignation and called it "regrettable," according to the South Korean presidential office.

Calls for Lee to resign began after South Korean tycoon Sung Woan-jong was found hanging from a tree in Seoul in an apparent suicide on April 9. Sung, who was under investigation for fraud and bribery, left a note listing names and amounts of cash given to top officials, including those who work for the President.

Lee and seven other politicians with links to the South Korean President are under investigation. cont...

South Korean PM offers resignation over bribery scandal

Suicide note leads to government bribery investigation

- Calls for Lee Wan-koo to resign began after South Korean tycoon Sung Woan-jong was found hanging from a tree in Seoul
- Sung, who was under investigation for fraud and bribery, left a note listing names and amounts of cash given to top officials

Figure 7: A CNN news article with story highlights and side information. The second block is the main body of the article. It comes with side information such as the title (first block) and the images with their captions (third block). The last block is the story highlights that assist in gathering information on the article quickly. These highlights are often used as the gold summary of the article in summarization literature.

Motivation Recent deep learning methods circumvent human-engineered features using continuous sentence features and still report results comparable to the state of the art without using any kind of linguistic annotation. Cheng and Lapata (2016) and Nallapati et al. (2016) used recurrent neural networks to read sequences of sentences to get a document representation which they use to label each sentence for extraction. However, these extractive methods often focus on the main body of the document from which sentences are extracted.

It is a challenging task to rely only on the main body of the document for extraction cues, as it requires document understanding. Documents in practice often have side information, such as title, image captions, videos, images and twitter handles, alongside the main body of the document. These types of side information are often available for newswire articles. Figure 7 shows an example of a newswire article. It shows the side information such as the title (first block) and the images with their captions (third block) along with the main body of the document (second block). The last block shows the manually written summary of the document in terms of "highlights" to allow readers to quickly gather information on stories. As one can see in this example,

gold highlights focus on sentences from the fourth paragraph, i.e., on the key events such as the "PM's resignation", "bribery scandal and its investigation", "suicide" and "leaving an important note". Interestingly, the essence of the article is explicitly or implicitly mentioned in the title and the image captions of the document.

Problem Formulation Given a document D consisting of a sequence of sentences $(s_1, s_2, ..., s_n)$ and a sequence of pieces of side information $(c_1, c_2, ..., c_p)$, we produce a summary S of D by selecting m sentences from D (where $m < n$). We judge each sentence s_i for its relevance in the summary and label it with $y_i \in 0, 1$ where 1 indicates that s_i should be considered for the summary and 0, otherwise. In this paper, we approach this problem in a supervised setting where we aim to maximize the likelihood of the set of labels $Y = (y_1, y_2, ..., y_n)$ given the input document D and model parameters θ:

$$P(Y|D;\theta) = \prod_{i}^{n} P(y_i|D;\theta) \tag{3}$$

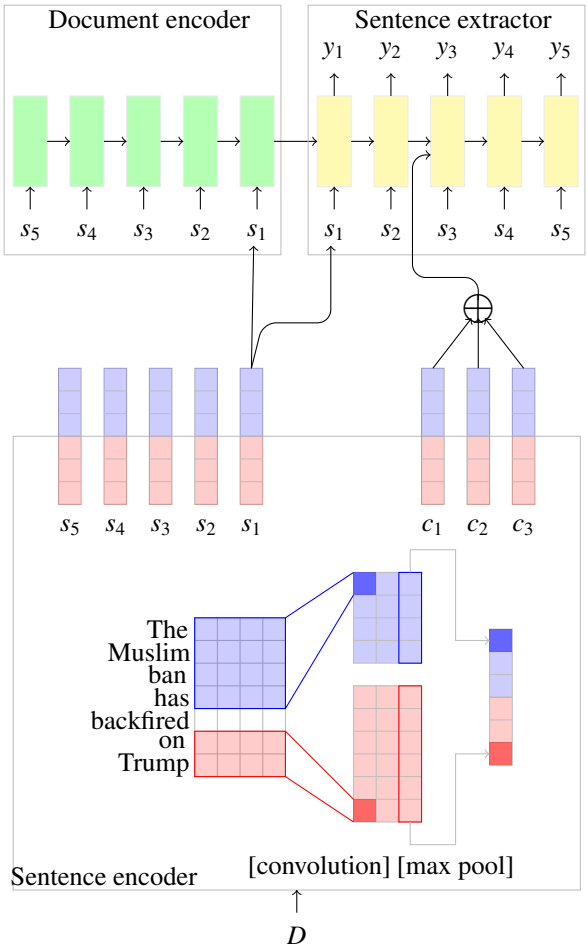

Figure 8: Hierarchical encoder-decoder model for extractive summarization with side information.

Model Figure 8 presents the layout of our model. We develop a general framework for single-document summarization with side information.

The main components of our model are a convolutional neural network sentence encoder (the bottom of Figure 8), a recurrent neural network document encoder (the top left of Figure 8) and an attention-based recurrent neural network sentence extractor (the top right of Figure 8). Our model exploits the compositionality of the document. It reflects the fact that a document is built of a meaningful sequence of sentences and each sentence is built of a meaningful sequence of words. With that in mind, we first obtain continuous representations of sentences by applying single-layer convolutional neural networks over sequences of word embeddings and then we rely on a recurrent neural network to compose sequences of sentences to get document embeddings. We model extractive summarization as a sequence labelling problem using a standard encoder-decoder architecture (Cho et al., 2014; Sutskever et al., 2014). First, the encoder reads the sequence of sentences $(s_1, s_2, ..., s_n)$ in D and then, the decoder generates a sequence of labels $(y_1, y_2, ..., y_n)$ labelling each sentence in D .

Our hierarchical document encoder resembles the architectures proposed by Cheng and Lapata (2016) and Nallapati et al. (2016), in that it derives the document meaning representation from its sentences and their constituent words. Like them, we also use recurrent neural networks to read the sequence of sentences in the document. However, our novel sentence extractor combines this document meaning representation with an attention mechanism (Bahdanau et al., 2014) over the side information to select sentences of the input document as the output summary. Our attention mechanism differs from both the standard attention mechanism (Bahdanau et al., 2014) which is used to locate the region of focus in the input, and the mechanism of Cheng and Lapata (2016) which directly extracts salient sentences after reading them. Instead, we use the attention mechanism to locate the region of focus in the side information.

Experimental Setup For our purposes, we used an augmented version of the CNN dataset (Hermann et al., 2015). This dataset has been used before for single-document summarization (Nallapati et al., 2016a; Cheng and Lapata, 2016; Nallapati et al., 2016b; Nallapati et al., 2016).

We augmented each article with the side information. We used a modified script of Hermann et al. (2015) to extract title and image captions, and we associated them with the corresponding articles. While all articles get associated with their titles, the availability of image captions varies from 0 to 414, with an average of 3 image captions per article. 40% of CNN articles have at least one image caption.

We need documents annotated with sentence importance information, i.e., each sentence in a document is labelled with 1 (summary-worthy) or 0 (not summary-worthy). We experimented with two types of such gold annotation: SentExtLabels and CollectiveOracle. SentExtLabels labels each sentence of the document in isolation for sentence extraction given the gold summary based on their semantic correspondence (Woodsend and Lapata, 2010). CollectiveOracle converts abstractive summaries to extractive ground truth by selecting a set of sentences from an article which collectively gives the highest ROUGE score with respect to the gold summary.

We use the standard splits of Hermann et al. (2015) for training, validation and test datasets. This divides our dataset into training, validation and test sets of sizes 90K, 1220 and 1093 documents respectively.

We report on both full length (three sentences with the top scores as the summary). Our decision of selecting three sentences is guided by the fact that there are 3.11 sentences on average in each gold highlight set in the training set.

Previous work has reported ROUGE-1 (R1) and ROUGE-2 (R2) scores to access informativeness,

and ROUGE-L (RL) to access fluency. In addition to R1, R2 and RL, we also report ROUGE-3 (R3) and ROUGE-4 (R4) capturing higher order *n*-grams overlap to assess informativeness and fluency simultaneously. We also complement our results with a human evaluation.

MODELS	R1	R2	R3	R4	RL	Avg.
LEAD	49.2	18.9	9.8	6.0	43.8	25.5
SEQ2SEQ						
SENTEXTLABELS	53.3	19.7	10.4	6.4	47.2	27.4
COLLECTIVEORACLE	54.3	21.1	11.1	6.9	48.9	28.5
SIDENET (COLLECTIVEORACLE)						
SIDENET+TITLE	55.0	21.6	11.7	7.5	48.9	28.9
SIDENET+CAPTION	55.3	21.3	11.4	7.2	49.0	28.8
SIDENET+FS	54.8	21.1	11.3	7.2	48.6	28.6
Combination Models (SIDENET+)						
TITLE+CAPTION	**55.4**	**21.8**	**11.8**	**7.5**	**49.2**	**29.2**
TITLE+FS	55.1	21.6	11.6	7.4	48.9	28.9
CAPTION+FS	55.3	21.5	11.5	7.3	49.0	28.9
TITLE+CAPTION+FS	55.4	21.5	11.6	7.4	49.1	29.0

Table 4: Ablation results on the validation set. We report recall scores of R1, R2, R3, R4, RL and their average (Avg.). LEAD is the baseline system selecting the "first" three sentences. SEQ2SEQ is a simple sequential encoder-decoder model which does not use any side information. We implement this by dropping out the attention mechanism in the sentence extractor. We experimented with two types of gold labels: SENTEXTLABELS and COLLECTIVEORACLE. SIDENET is our model. We experimented with three types of side information: title (TITLE), image captions (CAPTION) and the first sentence (FS) of the document. The bottom part of the table presents models with more than one type of side information. The best performing model (highlighted in boldface) is used on the test set.

MODELS	R1	R2	R3	R4	RL
LEAD	49.3	19.5	10.7	6.9	43.8
POINTERNET	51.7	19.7	10.6	6.6	45.7
SIDENET	**54.2**	**21.6**	**12.0**	**7.9**	**48.1**

Table 5: Rouge score for the full length summaries on the CNN test set. POINTERNET is the sentence extraction system of Cheng and Lapata.

Results Table 4 presents an ablation study on the CNN validation set. SEQ2SEQ with SENTEXTLABELS achieves scores of 53.3%, 19.7%, 10.4%, 6.4%, and 47.2% for R1, R2, R3, R4 and RL respectively. In comparison, SEQ2SEQ with COLLECTIVEORACLE achieves significantly better scores (54.3%, 21.1%, 11.1%, 6.9% and 48.9%). Following this, the rest of the models are trained with COLLECTIVEORACLE. Interestingly, all SIDENET models are superior to the LEAD baseline and it performs the best when TITLE and CAPTION are jointly used as side information (55.4%, 21.8%, 11.8%, 7.5%, and 49.2% for R1, R2, R3, R4, and RL respectively). It is better than the best SEQ2SEQ

model by 0.7 points on average, indicating that the side information is useful to identify the gist of the document. We evaluate this model on the test set.

Table 5 shows our final results. Our model (SIDENET) achieves state-of-the-art results by beating the model of Cheng and Lapata (2016) (POINTERNET) by 1.9 points for full length summaries, on average for all ROUGE scores.

Models	1st	2nd	3rd	4th
LEAD	0.15	0.17	0.47	0.21
POINTERNET	0.16	0.05	0.31	0.48
SIDENET	0.28	0.53	0.15	0.04
HUMAN	0.41	0.25	0.07	0.27

Table 6: Human evaluations: Ranking of various systems. Rank 1st is best and rank 4th, worst. Numbers show the percentage of times a system gets ranked at a certain position by human.

The results of our human evaluation study are shown in Table 6. We compare our SIDENET against LEAD, POINTERNET and HUMAN on how frequently each system gets ranked 1st, 2nd and so on, in terms of best-to-worst summaries. As one might imagine, HUMAN gets ranked 1st most of the time (41%). However, it is closely followed by SIDENET which is ranked 1st 28% of the time. In comparison, POINTERNET and LEAD were mostly ranked in 3rd and 4th places. We also carried out pairwise comparisons between all models in Table 6 for their statistical significance using a one-way ANOVA with post-hoc Tukey HSD tests with ($p < 0.01$). It showed that SIDENET is significantly better than LEAD and POINTERNET, and it does not differ significantly from HUMAN. On the other hand, POINTERNET does not differ significantly from LEAD and it differs significantly from both SIDENET and HUMAN. The human evaluation results reinforces our empirical results in Table 4 and Table 5 that SIDENET is better than LEAD and POINTERNET in producing informative and fluent summaries.

LEAD	• Seoul South korea's Prime Minister Lee Wan-koo offered to resign on monday amid a growing political scandal • Lee will stay in his official role until South Korean President Park Geun-hye accepts his resignation • He has transferred his role of chairing cabinet meetings to the deputy Prime Minister for the time being , according to his office
POINTERNET	• South Korea's Prime Minister Lee Wan-koo offered to resign on Monday amid a growing political scandal • Lee will stay in his official role until South Korean President Park Geun-hye accepts his resignation • Lee and seven other politicians with links to the South Korean President are under investigation
SIDENET	• South Korea's Prime Minister Lee Wan-Koo offered to resign on Monday amid a growing political scandal • Lee will stay in his official role until South Korean President Park Geun-hye accepts his resignation • Calls for Lee to resign began after South Korean tycoon Sung Woan-jong was found hanging from a tree in Seoul in an apparent suicide on April 9
HUMAN	• Calls for Lee Wan-koo to resign began after South Korean tycoon Sung Woan-jong was found hanging from a tree in Seoul • Sung, who was under investigation for fraud and bribery, left a note listing names and amounts of cash given to top officials

Figure 9: Summaries produced by various systems for the article shown in Figure 7.

In the end, Figure 9 shows output summaries from various systems for the article shown in Figure 7. As can be seen, both SIDENET and POINTERNET were able to select the most relevant sentence for

the summary from anywhere in the article, but SideNet is better skilled in producing summaries which are close to human authored summaries.

3.6.2 Abstractive Highlights Generation Oriented by Extractive Sentences

This section reports a method to generate abstractive highlights (which allows rewriting and paraphrasing the input text). This represents an additional challenge in relation to extractive summarization since it is necessary to generate coherent and cohesive text highlights.

The task of abstractive highlight generation has gained a lot of attention lately since it was considered a difficult problem where many deep learning systems could thrive. In particular, the works of Nallapati et al. (2016a), See et al. (2017) and Paulus et al. (2017) have approached this problem recently.

Our approach was motivated by the intuition that extractive systems can perform well in content selection while abstractive systems are better in generation. Accordingly, we built a pipeline solution where the sentences in the text are extracted by a coverage-based summariser and afterwards abstracted into highlights by a sequence-to-sequence deep learning model.

In training, we used the coverage-based extractive summariser + features reported in section 3.3, which was trained to select the best three sentences from the text.

After the sentences were extracted, each sentence was decoded by the abstractive sequence-to-sequence architecture in order to generate a highlight.

Our sequence-to-sequence architecture corresponds to the neural machine translation proposed by Bahdanau et al. (2014) where both the encoder and decoder consist of a stack of two bidirectional LSTMs (Hochreiter and Schmidhuber, 1997). The decoder has a global attention over the source hidden-states and a soft-max layer over target vocabulary to generate words.

For both the encoder and decoder we used the following parameters: the vocabulary was restricted to 50,000 words; we used 300 as the dimension of word embeddings; the embeddings were initialised with the Google News embeddings (Mikolov et al., 2013); we used 300 dimensional cells in each LSTM stack; we limited the size of input and output to 30 words. We used dropout rate of 0.3 in the last layer.

In order to train the neural network model, we constructed a dataset with source snippets from the article text and the target highlight from the summary highlight. For this, we used the same approach for extractive summarization, where, for each highlight, we selected the best sentences up to the maximum length of 30 words using the oracle approach proposed by Gillick et al. (2008). Further, each source and target pair was filtered by a criteria where the bigram overlap had to be bigger than a fixed threshold. This last condition aims to guarantee that the pairs must at least weakly entail.

Our model was trained using the OpenNMT framework Klein et al. (2017) with an early stop criterion based on the perplexity score in the development set.

As our target is to produce highlights, we used the CNN/Daily Mail corpus previously collected by Cheng and Lapata (2016). The CNN/Daily Mail is composed of texts from both the CNN.com and Dailymail.com websites. The training set consists of 83,568 articles from CNN and 193,986 from Daily Mail. The development set is composed by 1,220 articles from CNN and 12,147 from Daily Mail. The test set is composed by 1,093 articles from CNN and 10,350 from Daily Mail. For the test set, the average number of sentences per highlight summary is 1.51 on CNN and 3.03

on Daily Mail. The average tokens per highlights summary is 35.39 on CNN and 57.78 on Daily Mail.

Table 7 shows how our system scored using the ROUGE metric (Lin, 2004). We compare our approach with an extractive baseline which uses the coverage-based summariser with the restriction to select up to 50 words. We also compare our results with the results reported by Nallapati et al. (2016a) that also reported ROUGE values for the same dataset using an abstractive approach.

Table 7: Performance on CNN/Daily Mail test set using full-length Rouge-F1 metric. Bold faced numbers indicate best performing system.

Model	Rouge-1	Rouge-2	Rouge-L	Avg. size
Extractive	30.90	10.46	26.82	50
Nallapati et al. (2016a)	35.46	13.30	32.65	-
Our Method	**35.97**	**14.26**	**33.40**	44.63

The results show that our method was able to improve over the baseline and over the previous work of Nallapati et al. (2016a). Moreover, one of the advantages of our abstractive method against the extractive methods is the small average size for the summaries. To illustrate this, Figure 10 shows an example of extractive summary produced and its respective abstractive highlights.

EXTRACTIVE SUMMARY:

what do Walmart , Target , and now the Koch Brothers have in common with the American Civil Liberties Union , ColorOfChange.org , and the Center for American Progress ? all of them are adopting or advocating for hiring practices that open up work opportunities for people with convictions and leverage untapped potential in the labor market
Koch Industries ' recent announcement that it will " ban the box " – i.e. , remove from its job applications the check - box that asks about convictions – is a big step forward in the movement to break down barriers to employment for job - seekers with records
in a job market where employers that did n't previously do background checks now make them a routine part of hiring , qualified job - seekers are being screened out of the applicant pools for more and more jobs

---

ABSTRACTIVE HIGHLIGHTS:

- the Koch Brothers have in common with the American Civil Liberties Union , ColorOfChange.org , and the Center for American Progress ?
- Koch Industries said it will " ban the box "
- job - seekers are being screened out of the applicant pools for more and more jobs

Figure 10: Example of summary produced by our approach.

As mentioned, the abstractive summarization poses some extra difficulties for the process of highlight generation. One problem that needs to be addressed is the capacity of the model to identify pieces of information from different sentences in the original text. Another challenge is to leverage deep learning models with linguistic information, for example, semantic labels.

3.7 CCA-based summarization

This section describes another system which followed a different research direction. The main idea behind this system is the joint representation of documents and summaries in one low-dimensional space. Projection to that space is achieved by using the technique of Canonical Correlation Analysis (CCA; Hotelling (1935)). Sentences are selected for the final summary according to their proximity to the document in this space.

Background Canonical Correlation Analysis is a technique for multi-view dimensionality reduction, related to co-training (Yarowsky, 1995) (Blum and Mitchell, 1998). CCA operates on two vector spaces $\mathbb{R}^d$ and $\mathbb{R}^{d'}$ and finds a projection in a shared space $\mathbb{R}^m, m < min(d, d')$ so that the correlation between the two views is maximized at each coordinate and, at the same time, minimal redundancy between the coordinates is achieved. Canonical Correlation Analysis is a core technique for a set of proposed approaches for multi-view learning. In a multi-view setup, views can correspond to different modalities or different kinds of information about learning examples, and respective techniques learn a representation that captures all sources of information. In the context of document summarization, documents and summaries can be regarded as two different views of the same semantics that a short and a long text are meant to convey.

Model More specifically, given a set of documents and their summaries, documents are represented in a feature space and the summaries in a different feature space. By employing CCA, both spaces are projected to one shared low-dimensional space (as opposed to the original high-dimensional n-gram spaces). The projected space is a space where both documents and summaries are represented, in a way that document points and their corresponding summaries points are close, whereas document points are far enough from summaries that have nothing to do with them.

After training, where the projection is learned, extractive summarization proceeds as follows:

- for every new document, a feature vector is calculated in the original document space
- the document is split into sentences and for every sentence, a feature vector in the original summary space is calculated
- for every (document, sentence) tuple, projected vectors are calculated
- a proximity metric is calculated for every (document, sentence) projected vector tuple
- the extractive summary consists of the top n higher scoring sentences

Experiments and Results We experimented with this approach on the CNN/Daily Mail (Hermann et al., 2015), dataset, which consists of around 200,000 articles taken from the web editions of CNN and Daily Mail news websites. We have used n-gram (bigram) features for both the documents and the summaries, while the selection process is driven by calculating the cosine similarity between the projected vector of the document and the projected vector of each of the sentences. The dimensionality selected for the projection is 1500, whereas the output summary consists of the 3 top-scoring sentences (on average, every article of the dataset is accompanied by a three-sentence summary).

DM			
system	**R-1 R**	**R-2 R**	**R-L R**
LEAD-3	53.26	23.03	48.28
Cheng	56.00	**24.90**	**50.20**
CCA-text	56.55	22.67	35.74
CCA-mask	**57.02**	22.92	36.05

Table 8: Results two four different systems and our models on the Daily Mail part of the test corpus using the full length ROUGE metric. R-1 R stands for ROUGE-1 recall, R-2 R for ROUGE-2 recall and R-L R for ROUGE-L recall. CCA-text versions calculate ngrams over the input text, whereas CCA-mask versions calculate ngrams over a masked version of the text, where tokens are replaced by (part-of-speech, Brown cluster id) tuples.

Table 8 presents results for the the Daily Mail part of the corpus. Aligning with the literature on extractive summarization, we report ROUGE-1, ROUGE-2 and ROUGE-L recall scores (Lin, 2004), comparing with the LEAD-3 baseline (selecting the first three sentences of each article), Deep-Cls (Nallapati et al., 2016b) and NN-SE (Cheng and Lapata, 2016). Experimental results suggest that the proposed technique achieves promising results, while not being state-of-the-art.

Since this approach is inspired by a set of techniques proposed for multi-view learning, it is relatively easily extendable to account for more than the two views of documents and summaries. Examples for such views could be the titles of articles, the images that usually accompany news articles and the captions of the images. We are experimenting with more than two views and also with neural CCA variants (Deep Canonical Correlation Analysis (Andrew et al., 2013), Deep Generalised Canonical Correlation Analysis (Benton et al., 2017)) in order to further increase the performance of the system.

3.8 Final remarks and Future work

The main goal of this component is to generate storyline highlight summaries from the clusters created by the clustering component (WP3), in order to enable efficient news monitoring and content discovery, and thus satisfy the SUMMA BBC and DW use-cases. To this end, the summarization component can leverage the output of the semantic parser (from T5.1), and synthesise a coherent summary of all highlights that occur in the input story.

In order to accomplish this, initial work has been done in improving coverage-based summarization models with semantic graphs. In particular, both Propbank and AMR semantic formalisms were investigated. In connection with the actual state-of-art, additional work was reported in generating extractive and abstractive summaries using deep learning techniques. The results obtained by such models beat the current state-of-art, showing the efficiency of this type of approach to solve this problem.

In future work we intend to continue to investigate deep learning techniques and adapt them to the goal of storyline highlights generation. Also, we seek to experiment with new ways to incorporate semantic knowledge in the summarization task. Additionally, in order to leverage several summarization approaches currently being developed, we plan to experiment with the integration of the multi-document extractive summarizer and, the neural extractive and abstractive summarizers, from UEDIN and Priberam.

3.9 Publications

- Gruzitis, N., Gosko, D., and Barzdins, G. (2017). Rigotrio at semeval-2017 task 9: Combining machine learning and grammar engineering for amr parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 924–928, Vancouver, Canada. Association for Computational Linguistics

- Gruzitis, N. and Barzdins, G. (2016). The role of cnl and amr in scalable abstractive summarization for multilingual media monitoring. In *Controlled Natural Language*, volume 9767. Springer

- Narayan, S., Papasarantopoulos, N., Lapata, M., and Cohen, S. B. (2017b). Neural extractive summarization with side information. *CoRR*, abs/1704.04530

- Narayan, S., Gardent, C., Cohen, S. B., and Shimorina, A. (2017a). Split and rephrase. In *Proceedings of EMNLP*

- Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017). Creating training corpora for micro-planners. In *Proceedings of ACL*

- Narayan, S. and Cohen, S. B. (2016). Optimizing spectral learning for parsing. In *Proceedings of ACL*

- Osborne, D., Narayan, S., and Cohen, S. (2016). Encoding prior knowledge with eigenword embeddings. *Transactions of the Association for Computational Linguistics*, 4:417–430

- Narayan, S., Reddy, S., and Cohen, S. (2016). Paraphrase generation from Latent-Variable PCFGs for semantic parsing. In *Proceedings of INLG*

- Narayan, S. and Gardent, C. (2016). Unsupervised sentence simplification using deep semantics. In *Proceedings of INLG*

4 Sentiment Analysis of a Story

4.1 Original description of work

Task T5.3: Sentiment Analysis of a Story (Partners: PRIB, UEDIN) (Text taken from the proposal) The final component of this work package is a sentiment analyser for storylines. The news monitor will be able to see, for each story or a collection of events, a sentiment indicator. This sentiment indicator will aggregate the attitudes, expressed in social media sources monitored by the industrial partners, towards the set of events or story.

This task takes as input a story line, which consists of a set of news articles, blog posts, and short texts from micro-blogs, clustered together in Task 3.4. The main use of the semantic parser for the current task is to align text from social media as referring to specific instances of events in the articles. Once that is done, the sentiment score of a given event is calculated as the aggregate score of the sentiment of all social media references to that event, as in the above example.

Calculating the sentiment score of social media text is done separately from the alignment of social media text to news articles. As such, it can make use of standard techniques in sentiment analysis of social media, such as those described by Mohammad et al. (2013). From the machine learning perspective, these techniques usually bootstrap a basic lexicon for positive and negative sentiment, and then try to predict the sentiment based on this lexicon.

4.2 Summary of progress

This component aims to assign sentiment scores to automatically identified events, rather than a given document or sentence. The importance of this task is to help data analysts or media monitors to understand the aggregate sentiment on social media (e.g. Twitter) in respect to a particular story event (or highlight).

The key inputs for this component are the highlights produced in the summarization component of task T5.2. Therefore, in this first project phase, our efforts have been directed to the generation of quality highlights and on preparing a dataset to satisfy *SUMMA* evaluation requirements. Section 4.3 discusses how this dataset was constructed.

4.3 Annotation of a Dataset for Highlights and Tweets

The objective of creating an annotated dataset is to be able to evaluate both our summarization and sentiment analysis approaches in the context of *SUMMA* news data. In this annotation task, the leading partners were Priberam and UEDIN, with additional help from BBC and DW. Priberam provided semi-automatically clustered news articles and tweets for around 100 storylines. Priberam also provided a web system for the annotation process. UEDIN hired students to perform the annotation and helped with the annotation management. To help better understand the target concept of storyline and prune the automatically clustered news articles, Priberam's linguistic team, DW and BBC gave their feedback about the generated clusters in the annotation tool.

The annotation task was performed in three steps: database creation; highlights annotation; and tweets sentiment annotation. These steps are going to be described in further detail in the following sections.

4.3.1 Storylines and tweets collection

We created a storyline dataset by collecting news articles from BBC in the time span from 15th December 2016 and 14th January 2017. These articles were first clustered together forming storylines using the existing news clustering system from the *SUMMA* component T3.4, WP3. After that, a team of linguists from Priberam validated the clusters, excluding non-conformity clusters and filtering out misplaced articles. The resulting clusters were validated with BCC and DW partners to check the specificity level that the storylines should contain about their topic.

4.3.2 Storyline Highlights annotation

The objective of the task was to provide the main highlights and entities for a given cluster of stories (storyline). Each annotator was asked to write the highlights of the storyline, which as a whole can work as a summary. Second, the annotator was also asked to write the main entities present in the storyline. These were the requirements for the annotation of the highlights:

- Each highlight is a sentence that contains relevant information about the storyline.
- Each highlight should represent one and only one main aspect or event addressed by the storyline.
- Each highlight should be self-contained, i.e., it must be fully comprehensible independently of the other highlights.
- The highlights summary must be composed from 3 to 6 highlights.

Moreover, These were the requirements for the annotation of the entities:

- Entities could be interpreted as any important real-world object, such as persons, locations, organizations, products, etc.
- The annotated entities should be relevant to understand the storyline. They should play a main role in the news reported.
- It is not required to write all the entities present in the text, only the most relevant to the storyline.

The Figure 11 shows system built to annotate highlights and entities.

4.3.3 Tweet annotation

The objective of this task was to identify the sentiment of each related tweet towards each highlight (event from the text) and entity for a given cluster of stories (storyline), as annotated before. For each target, the annotator should mark if the sentiment of the tweet towards it is ‘not related’, ‘positive’, ‘neutral’ or ‘negative’. These were the requirements for the annotation of the entities:

- The annotator should ask the question: does the tweet make assumptions or take a position regarding the target?

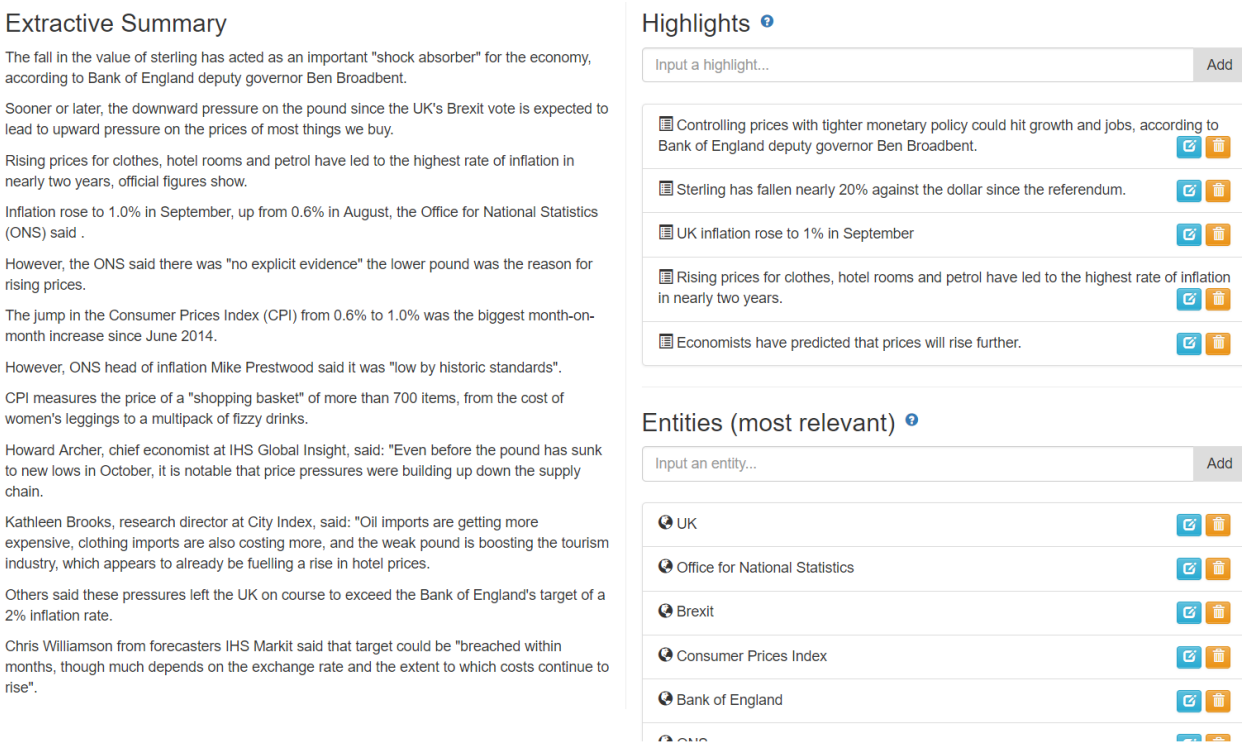

Figure 11: Highlights Annotation

- Based on the question above, if the answer is no, then the target must be marked as 'neutral', otherwise, it must be categorized into one of the two sentiment categories left (positive or negative).
- Each sentiment category translates the following idea:
  - Neutral: if the tweet has either no sentiment or the sentiment transferred to the target is unclear.
  - Positive: if the tweet transfers a cheerful, delighted excited, happy or in any way positive sentiment into the target.
  - Negative: if the tweet transfers a pessimistic, sad, disappointed, unfriendly or in any way negative sentiment into the target.
- If a tweet carries both positive and negative sentiment towards a target, the annotator may judge the predominant sentiment. In case of equal relevance, then mark as 'neutral'.

The Figure 12 shows system built to annotate tweets.

4.4 Final remarks and Future work

The final dataset produced by this effort will contain test data for evaluating the sentiment analysis and summarization tasks. This annotation guided us towards a better understanding of the problem of tweet classification and how to approach it in the future.

Our next step will be to build a system to automatically classify tweets based on the highlights, which will be facilitated by the fact that we now have an evaluation dataset suited for this task

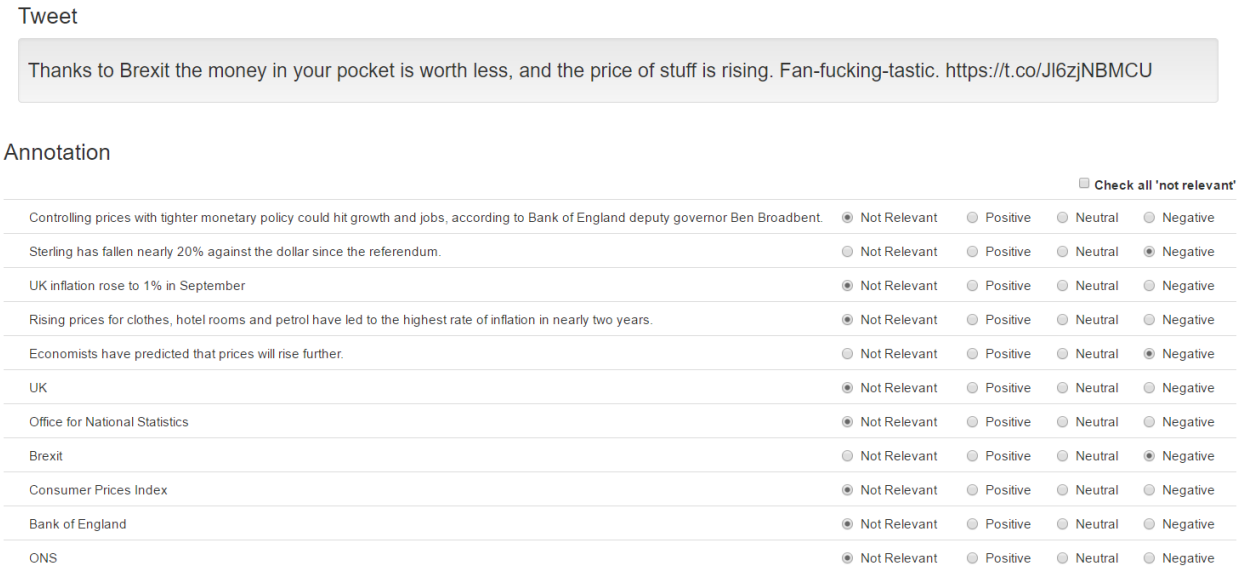

Figure 12: Tweets annotation

in *SUMMA*. Additionally, we'll also employ this dataset in the evaluation of the summarization approaches developed in task T5.2, WP5.

5 Conclusion

In this report we presented the current research and development undertaken in the SUMMA Natural Language Understanding work package, WP5. In particular, we presented our latest advances in semantic parsing, summarization and sentiment analysis components.

Regarding the first task (T5.1), we presented two working AMR parsers developed by the consortium: UEDIN's AMREager and LETA's CAMR wrapper, which show promising results in comparison to the literature. In particular, the multilingual AMR capabilities offered by AMREager enable novel possibilities both for the academic community and in the context of SUMMA. Additionally, we also discussed the performance improvements done in TurboParser (a PropBank semantic parser), by Priberam.

In the following task (T5.2), we discussed several different automatic summarization approaches carried out by UEDIN, Priberam and LETA. In the context of SUMMA, the objective of generating automatic summaries from news storylines (constructed in WP3) is to enable efficient news monitoring and content discovery, thus satisfying the SUMMA BBC and DW use-cases. Our current research efforts show promising results and there is still room to improve the current results in the remainder of the project.

Finally, in T5.3, we discussed our current efforts to create an evaluation dataset for summarization and sentiment analysis, which was a collaborative effort by Priberam, UEDIN, BBC and DW. Further in the project, we plan to use this dataset to evaluate both the developed sentiment analysis and summarization components.

All the different components being developed in T5.1, T5.2, and T5.3 are the result of our ongoing research effort to find the systems which better suit the use-cases of SUMMA. We plan to further experiment with these different approaches and eventually integrate the best components into a system which, given an input storyline, outputs highlight summaries with annotated sentiments.

References

Abend, O., Cohen, S. B., and Steedman, M. (2015). Lexical event ordering with an edge-factored model. In *HLT-NAACL*, pages 1161–1171.

Almeida, M. and Martins, A. (2013). Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 196–206, Sofia, Bulgaria. Association for Computational Linguistics.

Andrew, G., Arora, R., Bilmes, J. A., and Livescu, K. (2013). Deep canonical correlation analysis. In *ICML (3)*, pages 1247–1255.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation By Jointly Learning To Align and Translate. In *ICLR*, pages 1–15.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Barzdins, G. and Gosko, D. (2016). RIGA at SemEval-2016 Task 8: Impact of Smatch Extensions and Character-Level Neural Translation on AMR Parsing Accuracy. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1143–1147.

Benton, A., Khayrallah, H., Gujral, B., Reisinger, D., Zhang, S., and Arora, R. (2017). Deep generalized canonical correlation analysis. *CoRR*, abs/1702.02519.

Berg-Kirkpatrick, T., Gillick, D., and Klein, D. (2011). Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 481–490, Stroudsburg, PA, USA. Association for Computational Linguistics.

Blum, A. and Mitchell, T. (1998). Combining labelled and unlabelled data with co-training. In *Proc. of COLT*.

Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA. ACM.

Cheng, J. and Lapata, M. (2016). Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*.

Cohen, S. B. and Collins, M. (2012). Tensor decomposition for fast latent-variable PCFG parsing. In *Proceedings of NIPS*.

Cohen, S. B., Satta, G., and Collins, M. (2013). Approximate pcfg parsing using tensor decomposition. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 487–496, Atlanta, Georgia. Association for Computational Linguistics.

Damonte, M., Cohen, S. B., and Satta, G. (2017). An Incremental Parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain, April 3-7, 2017. Association for Computational Linguistics.

Das, D., Chen, D., Martins, A. F., Schneider, N., and Smith, N. A. (2014). Frame-semantic parsing. *Computational linguistics*, 40(1):9–56.

Filatova, E. and Hatzivassiloglou, V. (2004). A formal model for information selection in multi-sentence text extraction. In *Proceedings of Coling 2004*, pages 397–403, Geneva, Switzerland. COLING.

Flanigan, J., Dyer, C., Smith, N. A., and Carbonell, J. (2016). Generation from abstract meaning representation using tree transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California. Association for Computational Linguistics.

Flanigan, J., Thomson, S., Carbonell, J., Dyer, C., and Smith, N. A. (2014). A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Ganchev, K. and Das, D. (2013). Cross-lingual discriminative learning of sequence models with posterior regularization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1996–2006, Seattle, Washington, USA. Association for Computational Linguistics.

Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017). Creating training corpora for micro-planners. In *Proceedings of ACL*.

Gillick, D., Favre, B., and Hakkani-Tur, D. (2008). The icsi summarization system at tac 2008. In *Proc. of Text Understanding Conference*.

Greenbacker, C. F. (2011). Towards a framework for abstractive summarization of multimodal documents. In *Proceedings of the ACL 2011 Student Session*, HLT-SS '11, pages 75–80, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gruzitis, N. and Barzdins, G. (2016). The role of cnl and amr in scalable abstractive summarization for multilingual media monitoring. In *Controlled Natural Language*, volume 9767. Springer.

Gruzitis, N., Gosko, D., and Barzdins, G. (2017). Rigotrio at semeval-2017 task 9: Combining machine learning and grammar engineering for amr parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 924–928, Vancouver, Canada. Association for Computational Linguistics.

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. *ArXiv e-prints*, abs/1506.03340.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hotelling, H. (1935). Canonical correlation analysis (cca). *Journal of Educational Psychology*.

Kingsbury, P., Palmer, M., and Marcus, M. (2002). Adding semantic annotation to the penn treebank. In *Proceedings of HLT-02*, San Diego.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Lampouras, G. and Vlachos, A. (2017). Sheffield at semeval-2017 task 9: Transition-based language generation from amr. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 586–591, Vancouver, Canada. Association for Computational Linguistics.

Lewis, M. and Steedman, M. (2013). Combining distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In Marie-Francine Moens, S. S., editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Martins, A., Figueiredo, M., Aguiar, P., Smith, N., and Xing, E. (2011a). An augmented lagrangian approach to constrained map inference. In Getoor, L. and Scheffer, T., editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 169–176, New York, NY, USA. ACM.

Martins, A. F. and Almeida, M. S. (2014). Priberam: A turbo semantic parser with second order features. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 471–476.

Martins, A. F., Figueiredo, M. A., Aguiar, P. M., Smith, N. A., and Xing, E. P. (2015). Ad 3: Alternating directions dual decomposition for map inference in graphical models. *The Journal of Machine Learning Research*, 16(1):495–545.

Martins, A. F., Smith, N. A., Aguiar, P. M., and Figueiredo, M. A. (2011b). Structured sparsity in structured prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1500–1511. Association for Computational Linguistics.

Martins, A. F. T., Almeida, M. B., and Smith, N. A. (2013). Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

May, J. (2016). Semeval-2016 task 8: Meaning representation parsing. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1063–1073. Association for Computational Linguistics.

May, J. and Priyadarshi, J. (2017). Semeval-2017 task 9: Abstract meaning representation parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 536–545, Vancouver, Canada. Association for Computational Linguistics.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 557–564, Berlin, Heidelberg. Springer-Verlag.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nallapati, R., Zhai, F., and Zhou, B. (2016). SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents. *ArXiv e-prints*.

Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., and Xiang, B. (2016a). Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Nallapati, R., Zhou, B., and Ma, M. (2016b). Classify or select: Neural architectures for extractive document summarization. *ArXiv e-prints*.

Napoles, C., Gormley, M., and Van Durme, B. (2012). Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

Narayan, S. and Cohen, S. B. (2016). Optimizing spectral learning for parsing. In *Proceedings of ACL*.

Narayan, S. and Gardent, C. (2016). Unsupervised sentence simplification using deep semantics. In *Proceedings of INLG*.

Narayan, S., Gardent, C., Cohen, S. B., and Shimorina, A. (2017a). Split and rephrase. In *Proceedings of EMNLP*.

Narayan, S., Papasarantopoulos, N., Lapata, M., and Cohen, S. B. (2017b). Neural extractive summarization with side information. *CoRR*, abs/1704.04530.

Narayan, S., Reddy, S., and Cohen, S. (2016). Paraphrase generation from Latent-Variable PCFGs for semantic parsing. In *Proceedings of INLG*.

Nenkova, A., Passonneau, R., and McKeown, K. (2007). The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2).

Osborne, D., Narayan, S., and Cohen, S. (2016). Encoding prior knowledge with eigenword embeddings. *Transactions of the Association for Computational Linguistics*, 4:417–430.

Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization. *ArXiv e-prints*.

Ranta, A. (2011). *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *ArXiv e-prints*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of NIPS*.

Wang, C., Xue, N., and Pradhan, S. (2015). A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375, Denver, Colorado. Association for Computational Linguistics.

Woodsend, K. and Lapata, M. (2010). Automatic generation of story highlights. In Carberry, S. and Clark, S., editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 565–574, Uppsala, Sweden. Association for Computational Linguistics.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL*.

Yih, W.-t., Goodman, J., Vanderwende, L., and Suzuki, H. (2007). Multi-document summarization by maximizing informative content-words. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 1776–1782, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

ENDPAGE

SUMMA

H2020-ICT-2015 688139

D5.1 Initial Progress Report on Natural Language Understanding