Multilingual Natural Language Generation within Abstractive Summarization

Simon Mille[1], **Miguel Ballesteros**[1], **Alicia Burga**[1], **Gerard Casamayor**[1], and **Leo Wanner**[1,2]

Abstract. With the tremendous amount of textual data available in the Internet, techniques for abstractive text summarization become increasingly appreciated. In this paper, we present work in progress that tackles the problem of multilingual text summarization using semantic representations. Our system is based on abstract linguistic structures obtained from an analysis pipeline of disambiguation, syntactic and semantic parsing tools. The resulting structures are stored in a semantic repository, from which a text planning component produces content plans that go through a multilingual generation pipeline that produces texts in English, Spanish, French, or German. In this paper we focus on the lingusitic components of the summarizer, both analysis and generation.

1 Introduction

With the tremendous amount of multilingual textual data available in the Internet, techniques for intelligent abstractive text summarization in the language of the preference of the user enjoy a steadily increasing demand for different applications, among them journalism and media monitoring. Thus, journalists and media monitors have to review a large number of press articles on a daily basis, a considerable number of which may not be available in their native language. We present work in progress that tackles the problem of multilingual text summarization using semantic representations.

The most popular summarization strategy is still "extraction"-oriented. Text fragments (in general, entire sentences, but in some cases also phrases), are selected from one or more source documents, based on some relevance metric, and the most relevant fragments are put together in a summary (see, e.g., [19] for an overview). Although extractive summarization can be addressed with little linguistic analysis and, in the case of sentence-based selection metrics, the resulting summaries are always grammatically correct, it is known to have some significant shortcomings. For instance, the selection of the content to be included into the summary is rather coarse-grained and surface-(instead of knowledge-)oriented, and the summaries tend to lack internal coherence between the selected text fragments. Furthermore, in general, the summaries are monolingual, i.e., the original text and the summary are in the same language.

Opposed to extractive summarization is "abstractive summarization". Abstractive (i.e., concept-based) summarization analyzes the original textual material using language parsing and/or Information Extraction into intermediate linguistic or conceptual representations. Content selection relevance-driven techniques are then applied to these representations to choose the content elements that are to be communicated in the summary. From the chosen content elements, a summary is generated using deep Natural Language Generation (NLG) techniques. A number of approaches to abstractive summarization have been proposed. Some attempt to adapt extractive techniques to abstractive summarization [23]. Others do not use abstract representations and remain at a superficial level [15, 26], or use partially abstract structures, be it because not all the content of the input text is represented [20, 17], or because some idiosyncratic features are maintained [37]. It is also not always the case that deep generation is used. For instance, Genest and Lapalme [17] and Saggion and Lapalme [35] start from templates, Ganesan et al. [15] from word lattices, and Cheung and Penn [8] and Genest and Lapalme [16] from syntactic structures. Liu et al. [22] do not have a proper generation component at all. Liu et al. [22] and Cheung and Penn [8] apply sentence fusion, rather than content selection.

We developed techniques for abstractive summarization that are capable of generating multilingual summaries in response to a user query on a specific content element using full state-of-the-art (deep) language analysis and language generation mechanisms, combining statistic and rule-based techniques. In this paper, we focus on the general architecture of the summarizer and its generation module.

2 An architecture for abstractive summarization

2.1 Theoretical framework

The theoretical framework that underlies our system is the Meaning-Text Theory [27]. MTT is based on the notion of dependency, which establishes a relation of "governance" between two elements.

The MTT model supports high expressiveness at the three main levels of the linguistic description of written language: semantics, syntax and morphology, while facilitating a coherent transition between them via intermediate levels of deep syntax and deep morphology. In total, the model foresees five strata; at each stratum, a clearly defined type of linguistic phenomena is described in terms of distinct dependency structures.

- **Semantic Structures** (SemSs) are predicate-argument structures in which the relations between predicates and their arguments are numbered in accordance with the order of the arguments.
- **Deep-syntactic structures** (DSyntSs) are dependency trees, with the nodes labeled by meaningful ("deep") lexical units (LUs) and the edges by actant relations *I*, *II*, *III*, ..., *VI* (in accordance with the syntactic valency pattern of the governing LU) or one of the following three non-argumental relations: *ATTR*(ibute), *COORD*(ination), *APPEND*(itive).
- **Surface-Syntactic Structures** (SSyntSs) are dependency trees in which the nodes are labeled by open or closed class lexemes and the edges by grammatical function relations of the type *subject*, *oblique_object*, *adverbial*, *modifier*, etc.

[1] TALN, Pompeu Fabra University, C/ Roc Boronat, 138, 08018 Barcelona, email: simon.mille| miguel.ballesteros|alicia.burga|gerard.casamayor| leo.wanner@upf.edu
[2] ICREA, Passeig Lluís Companys, 23, 08010 Barcelona

- **Deep-Morphological Structures** (DMorphSs) are chains of lexemes in their base form (with inflectional and PoS features being associated to them in terms of attribute-feature pairs) between which a precedence relation is defined and which are grouped in terms of constituents.
- **Surface-Morphological Structures** (SMorphSs) are chains of inflected word forms, i.e., sentences as they appear in the corpus, except that orthographic contractions still did not take place.

The analysis and generation modules in our abstractive pipeline draw upon these strata. In particular, the tasks of language analysis and language generation can be seen as a sequence of mappings between adjacent strata; for analysis, starting from text and arriving at a semantic (or conceptual) representation , and for generation, starting from a semantic representation up to the text surface.

2.2 A pipeline for abstractive summarization

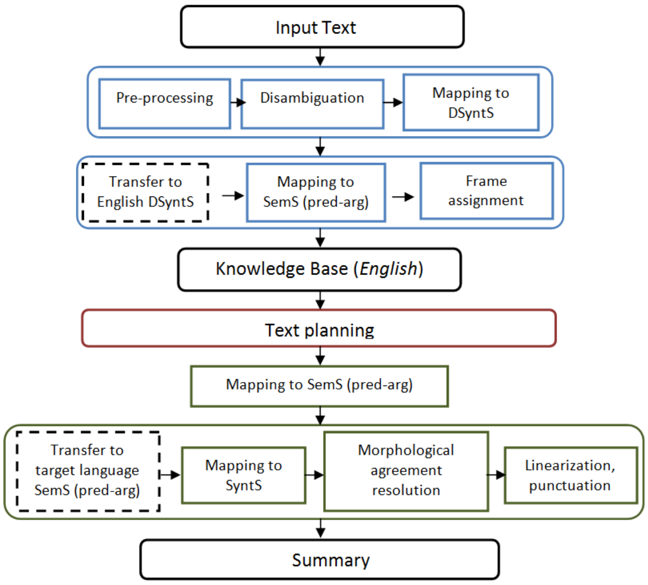

Figure 1. General architecture of the summarizer: Analysis, text planning, and generation

The implementation of our abstractive summarizer is based on a sequence of modules that realize the sequence of transitions between the different strata of the MTT model. The pipeline shown in Figure 1 can be divided into three main parts:

1. **Language analysis**: Language analysis is carried out by a text analysis pipeline that takes as input the textual content of a document in a given language. This content is first analyzed and represented as a forest of DSyntSs. In the case that the input language is different from English, every lexeme in the DSyntSs is mapped onto an English lexeme using bilingual dictionaries in order to arrive at a kind of *interlingua* structure that facilitates language-neutral representations (see Subsection 3.3 for a justification). These English "interlingua" structures are then mapped onto semantic structures, enriched with Frames from the FrameNet lexicon [13], modeled as RDF triples, and stored in a semantic repository.
2. **Text planning**: Conceptual summarization is approached by assessing the relevance of the semantic structures produced by the language analysis step in relation to a specific entity which constitutes a topic of interest for the end user and to which the generated summary is tailored. In addition to determining the relevance of contents, our text planning component also attempts to guarantee a degree of coherence in the summary to be generated by sorting relevant contents in a sequence that satisfies certain choerence constraints, e.g. grouping together in the text contents making reference to the same entities. Relevance calculations are based on relative cooccurrence metrics of word senses and references to entities detected in the original documents during language analysis, the cooccurrence metrics being obtained from pre-existing corpora of annotated documents.
3. **Natural language generation**: Following this planning step, linguistic generation starts by transferring the lexemes associated to the semantic structures to the desired target language, using available multilingual lexical resources. Then, the structure of the sentence is determined and all grammatical words are introduced and linked with syntactic relations. Finally, all morphological agreements between the words are resolved, the words are ordered and punctuation signs are introduced.

3 Language analysis

3.1 Tokenization and disambiguation

Language analysis starts by determining sentence and token boundaries using Bohnet et al.'s [6] tools. Rather than addressing tokenization at word level, however, our analysis pipeline treats each sequence of words referring to a specific entity as an atomic unit of meaning. In doing so, we seek to avoid unnecessary internal analysis of multiword expressions which may not even have a strictly compositional meaning (as, e.g., *United States of America*), and also to eventually obtain predicate-argument structures in which the arguments are not just words, but expressions with an atomic meaning.

To determine the disambiguated senses of individual words and the entities referred to by single words or phrases, we use Babelfy[3] [29]. Babelfy addresses both Word Sense Disambiguation and Entity Linking against BabelNet [30], a large multilingual semantic network organized around *Babel synsets* resulting from mapping WordNet synsets and Wikipedia pages. The large coverage of BabelNet allows Babelfy to annotate both Named Entities and conceptual meanings. All multiword expressions annotated by Babelfy are considered by the following modules as a single token.

3.2 Deep-syntactic parsing

Once the texts are clean, tokenized and the words are disambiguated against BabelNet, they are sent to a parsing module that carries out in sequence Surface-Syntactic and Deep-Syntactic Parsing.

For **Surface-Syntactic Parsing**, we use Bohnet et al.'s [6] joint lemmatizer, part of speech tagger, morphology tagger and dependency parser, which follows a transition-based approach with beam-search. Trained on a surface syntactic treebank, the joint parser produces the surface syntactic tree for an unseen sentence.[4]

For **Deep-Syntactic Parsing**, we use a SSynt-DSynt transducer. The objective of the transducer is to identify and remove all functional words (auxiliaries, determiners, void prepositions and conjunctions) in the surface-syntactic tree and to generalize the syntactic dependencies obtained during the previous stage, while adding subcategorization information for lexical predicates. Two different transducers have been developed. One is based on a statistical model

[3] www.babelfy.org
[4] The details on Bohnet et al.'s system can be obtained from the original work. It suffices to note here that it produces very competitive scores for all the tasks it performs, for a wide range of languages.

and the other is rule-based. The statistical transducer (see [1, 2] for details) is trained on parallel SSynt and DSynt corpora (see for instance [24] for an example in Spanish). The DSyntS-SSyntS transducer has the potential to be trained for any language in which there are parallel DSynt and SSynt available treebanks. Currently, it is the case for English and Spanish. The rule-based transducer is implemented a graph-transduction grammars that have access to language-specific lexicons to remove the void prepositions and conjunctions [28], when any is available. The rule-based version is available for English, Spanish, German, and French.

3.3 Mapping to abstract representations and frame assignment

For mapping deep-syntactic structures to more abstract linguistic representations, large-scale lexical resources are needed. Unfortunately, such resources are available, at this point, only for English; see, e.g., PropBank [21], FrameNet [13], VerbNet [36], and the mappings between them (SemLink [32]). For this reason, we chose to map all input languages to English.

After the SSynt-DSynt transduction, the obtained structure does not contain any functional words, which tend to be idiosyncratic. The nodes are labeled with meaningful lexemes.[5] Using multilingual resources such as BabelNet (see Sections 3.1 and 5.1), it is possible to obtain the translations of these words into English. Once this is done, the combination of the subcategorization information in the deep-syntactic structure and SemLink allows us to obtain Frame annotations on top of connected predicate-argument structures. The latter follow the principles of the Meaning-Text Theory model, with the addition of a subset of relations such as *Location*, *Time*, etc., which facilitate the further processing. During this step, shared argumental positions are made explicit and idiosyncratic structuring such as the representation of raising and control verbs is generalized.

4 Text Planning: Planning the Summary

Before a summary can be generated, it is necessary to determine, on the one hand, the content that is to be communicated to the user and, on the other hand, the discourse structure of the determined content. These two tasks are commonly referred to in NLG literature as *text* or *document planning* [34].

The production of summaries in our system assumes that summaries are generated in response to a user query in which an entity in the semantic repository is specified. The entity must correspond to a BabelNet synset, identified by any of the multilingual lexicalizations associated to it in BabelNet. If multiple synsets match the string introduced by the user, the user is asked to choose from the available meanings. The summary to be produced should contain content in the semantic repository that is relevant to the queried entity. The task of the text planning module is to determine the set of most relevant content elements and generate an ordered list out of them.

4.1 Ranking semantic structures

Following previous graph-based approaches to text planning [31, 10], our approach adopts a graph view on the content of the semantic repository where nodes correspond to predicate-argument structures produced by the analysis pipeline, and edges indicate entity-sharing relations between nodes. For each query, a *query graph* is created that contains as nodes all predicates that have the user-specified entity as one of its arguments. This initial set is extended recursively with other nodes that share at least one argument with relations already in the graph, up to a fixed depth. The resulting graph serves to constrain the planning task to a set of related contents, in a similar fashion to past works such as [25, 9, 7].

Given a query graph, we formulate the planning of summaries as a *ranking problem*, similar in spirit to other text planning implementations based on ranking contents [9, 7, 14]. In our ranking formulation all nodes in a query graph must be ranked according to some function that indicates their relevance. Consequently, the text plans produced by our method are sorted lists of nodes in a query graph. The ranking starts from an initial distribution of relevance obtained from co-occurrence counts in a corpus of texts analyzed with Babelfy. For each entity annotated in the corpus, we thus estimate its probability of being annotated in the same document as any other entity. Predicate-argument structures are then assigned an initial rank according to the probabilities of their arguments co-occurring with the queried entity.

Some predicates may have none of their arguments annotated in the Babelfy corpus. In this case, we have no empirical basis to assess their relevance. In order to ameliorate this situation, we distribute the relevance from those nodes that do have some probability assigned to them to their neighboring nodes in the query graph. This is achieved by iteratively multiplying the initial distribution of relevance to nodes of the query graph with an adjacency matrix of the query graph which has been modified so that it can be interpreted as a Markov Chain. That is, given a node, its transition probabilities are calculated from the relevance scores of its target nodes and normalized according to the sum of relevance of all nodes reachable from the initial node. This procedure, which is similar to web ranking [11], produces a near-stationary distribution of relevance in which the initial relevance scores have been adjusted according to the graph topology.

4.2 Producing a coherent text plan

As pointed out above, the goal of the text planning module is not only to determine the relevance of content elements with respect to the user query, but also to define a discourse structure for these elements, i.e., an ordering of the elements that enforces a certain degree of coherence in the resulting text. We do so by ensuring that new predicate-argument structures are added to a text plan only if they are semantically related to the content elements already in the plan. More precisely, we guarantee entity-coherence by performing a *graph exploration* of the query graph, which consists in visiting only those nodes that are connected to nodes that have been already visited. This notion of entity-based coherence is inspired by theories of local coherence such as the Centering Theory [18, 33].

Since the edges in the content graph capture argument-sharing relations between predicates, the sequence of visited nodes is such that, for every node, at least one of its arguments is either the requested entity, or an argument that has already been introduced into the plan. The traversal of the query graph is done in a greedy way. Starting from the set of predicates that have the queried entity as their argument, the most relevant node from all those that available is always selected. Once a node is selected, all the nodes in the query graph connected to it become available for selection. The traversal of the graph produces the ordered sequence that constitutes the input of the surface generation pipeline.

[5] This assumption is not entirely true since our DSyntSs still contain, e.g., support verbs (as *deliver* in *John delivered his first speech in the Congress*), which are generally assumed to be void of meaning as well. In genuine MTT DSyntSs, support verbs do not appear as such either. However, we think that this simplification can be tolerated without a too significant loss of quality.

5 Multilingual text generation

The predicate-argument structures produced by the text planning module are obtained from translating words from the source documents into English (see Section 3.3). In order to generate multilingual text, however, it is necessary to map them to linguistic structures that serve as a starting point for multilingual linguistic generation, which, in turn, requires language-specific lexical resources that capture the lexical and syntactic characteristics of each language. In the next subsection, the creation of such multilingual lexical resources is explained in detail.

5.1 Multilingual lexical resources

For multilingual generation, we need to create lexicons for each language we cover. These lexicons must not only contain language-specific vocabulary, but also be linked to our pivot language, namely English. Given that BabelNet senses annotated during the analysis stage are language-independent, we use them as the cross-linguistic link. Below, we detail the creation procedure and structure of the language-specific lexicons used to go from predicate-argument structures and BabelNet synsets to each language.

The languages supported by our multilingual generation pipeline (English, French, Spanish and German) have a satisfactory amount of NLP resources. The experimental compilation of the corresponding language-specific lexicons was done in different stages. First of all, three texts in each language were randomly selected. Thus, a set of eight texts (around 2,400 tokens) was used as base for the language-specific lexicons.[6] Given that word sense ambiguity is a problem inherent to any language, it was necessary to disambiguate and recognize the right sense of a lexical unit before assigning any specific BabelNet id to it. Babelfy, which as explained in Section 3.1, is connected to BabelNet, was used for disambiguation, using the API offered to remotely access the service. As output of this step, a list of unique BabelNet ids (1,013 items in total) was obtained, which served as the basis for creating the lexicons. This list has then been locally enriched with the word form linked to each id in each language. Using this list as base, for each LU, its part of speech, its lemma, its BabelNet id and its government pattern, i.e., its subcategorization frame, are stored. Within the government pattern, the information collected for each argument includes its part of speech, the preposition introducing it (if it is required by the described LU) and the corresponding case. Below, the entries for the same specific BabelNet id in German (a language with case) and in Spanish are shown.

SPANISH	GERMAN
"contar_VV_01":_verb_ {	"sagen_VV_01":_verb_ {
lemma = "contar"	lemma = "sagen"
bn = bn:00091011v	bn = bn:00091011v
gp = {	gp = {
I = {dpos = "N"}	I = {dpos = "N" case = "nom"}
II = {dpos = "N"}	II = {dpos = "N" case = "acc"}
III = {dpos = "N" prep = "a"}}}	III = {dpos = "N" case = "dat"}}}

From the English structure, the system thus turns to the lexicons to obtain information about the specific characteristics of the sentences to be generated in each language. If no specific information is added, the system interprets that there are no restrictions with respect to the argument in question. Thus, the four compiled parallel language-specific lexicons serve in a direct way for the multilingual generation pipeline, allowing the mapping from English to any of the other languages involved. Potentially, the mapping could be even done not only from English to other language, but from any other language included in the system to each other.

[6] Although it can be argued that the work is based on a small sample of vocabulary, the sample is big enough to test the adopted methodology.

5.2 Hybrid NLG system

The lexical resources described in Section 5.1 are meant to be used together with generation grammars, which are rule systems that produce successively the different layers of representation mentioned in Section 2. In this section, we describe the different submodules of the NLG pipeline, together with their alternative Machine Learning implementations. In order to understand better the process, Figure 2 includes some intermediate structures of this pipeline.

1. Mapping to output language predicate-argument structures
Starting from the structures provided by the text planning module (see Section 4), first, some idiosyncratic transformations are made to adjust the structures to the predicate-argument format understood by our generation pipeline, and then, the English labels of the nodes are translated into the desired target language using the lexicons detailed in Section 5.1.

2. Mapping to syntactic structures
Once genuine predicate-argument structures in the target language are available, the first task is to find which node in each structure is most likely to be the root of the dependency tree. That is, we want to identify what will be the main verb of the sentence, or the word that triggers its appearance. The main node is typically a word (i) that is predicate, (ii) that has more participants than any other predicate of the structure, and (iii) that is not involved in a semantic relation of secondary relevance. Adjectives, adverbs, prepositions and nouns are possible alternatives to verbs when no verb is available. Around the main node, the deep-syntacticization module builds the rest of the syntactic structure of the sentence. In particular, it is able to decide if a main predicate has to be introduced, or what will be realized as an argument, an attribute, or a coordination.

The procedure of the retrieval of deep-syntactic target structures has been successfully tested on around 39,000 sentences: more than 99% of the semantic structures are mapped to well-formed deep-syntactic structures. In the rest of the cases, the generator is unable to produce any syntactic tree and a fallback message is returned.

The next step in the procedure is to obtain surface-syntactic structures, i.e., to generate all functional words and labeling the dependencies with SSynt relations. In the same fashion as for SSynt–DSynt transduction in the case of analysis, we use two alternative approaches for DSynt–SSynt transduction in the case of generation. For languages with limited amount of annotated data (as, e.g., French or German), a rule-based system is preferred, but if multilayered corpora of reasonable size are available (as Spanish and English), training statistical tools is also possible.

For rule-based transduction, we use an adapted version of the MARQUIS generator [38]. MARQUIS had been designed for data-to-text generation. It starts from air quality and meteorology time series, and uses language-specific resources that contain a fine-grained description of all the concepts and words in the air quality domain. Generation in the context of abstractive summarization is a case of text-to-text generation. That is, we cannot focus on the concepts of a specific domain. Rather, any concept can be present in a semantic structure, and there are no lexical resources that are complete enough to contain all of them. As a consequence, MARQUIS's graph-transduction grammars had to be adapted.

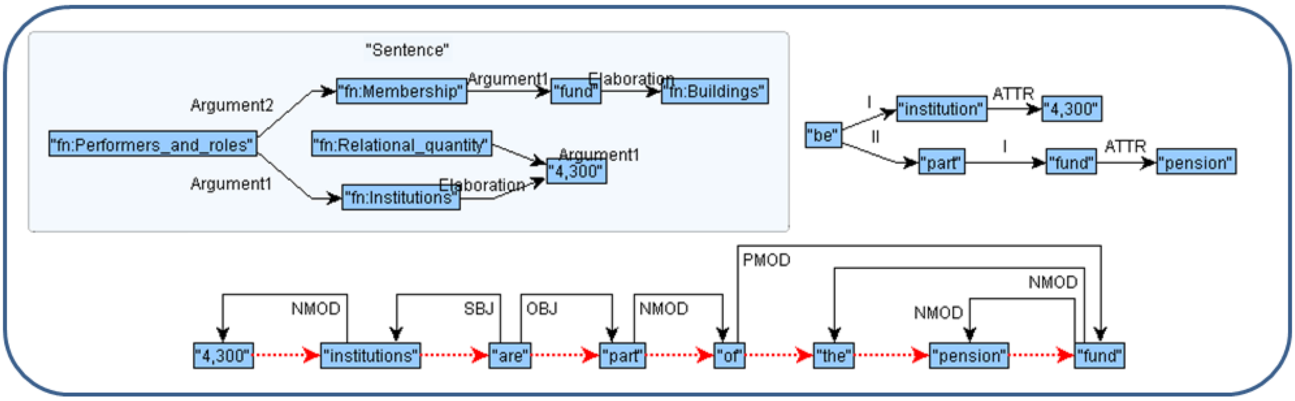

Figure 2. Sample text plan (top left), deep-syntactic structure (top right) and surface-syntactic + morphologic structures (bottom)

For machine learning-based transduction, we developed a series of Support Vector Machine-based transducers; cf., [3] for details.

3. Morphological agreement resolution and surface form retrieval

During the generation of syntactic structures, morphological features of individual words are already inserted (e.g., nominative case for a German subject). During the transition to the morphological structure, agreement is established (using the introduced morphological features and the fine-grained syntactic relations in the SSyntSs) and surface forms of the words are retrieved using a full-form dictionary.

In order to obtain the full-form dictionary, we run the morphological tagger of our surface syntactic parser on a large collection of texts and store each possible combination of surface form, lemma and morphological features. We can therefore retrieve a surface form given a lemma and a set of morphological features. The size of the text collection is crucial in order to ensure a large coverage. For instance, for English, we use the entire Gigaword corpus.[7]

4. Linearization of Unordered Syntactic Dependency Trees

The linearization of unordered syntactic dependency trees, i.e., word order determination, is performed with the state-of-the-art Bohnet et al. [5] linearizer. This linearizer is trained on a surface-syntactic treebank. It produces a statistical model that is capable of determining the word order in a sentence by using mainly surface-syntactic relations and part-of-speech tags.

6 Conclusions and Future Work

In this paper, we presented work in progress for the production of abstractive summaries about specific entities from contents obtained from the analysis of multiple texts. We have covered the resources, tools and techniques applied to obtain the summaries, placing special emphasis on text planning and the multilingual generation component.

In the future, we plan to evaluate the pipeline in one or more domains and use the results to determine what components require improvement. Our first application domain will be the production of multilingual summaries from news articles in the scope of the MULTISENSOR project [8]. We expect our natural language processing tools to perform better in journalistic texts than in more specialized domains that may require models obtained from domain-specific corpora. Crucially, the entities and concepts found in press articles are also more likely to be covered by BabelNet, which plays a crucial role in deciding what contents go into the summary. Specialized domains, e.g. medical or legal texts, may use terminology and make reference to entities only found in specialized kwnowledge and lexical resources. Creating multilingual resources and tools for specific domains is one of the major limitations of applying an abstractive approach to summarization.

The evaluation of our approach will involve both a quantitative evaluation where system-produced summaries are compared to a gold standard of manually written (abstractive) summaries, and a qualitative evaluation in which users will be handed a questionnaire designed at reviewing various facets of the texts: relevance, coherence, grammaticality, readability, etc. Considering the pipeline architecture of our system and the problems introduced by error-propagation, an individual evaluation of each module will also be conducted to identify the most problematic areas. We are particularly interested in finding ways to cope with noisy output from the text analysis component during text planning and linguistic generation, in order to avoid generating ungrammatical or meaningless sentences.

With respect to the lexicons used in the surface generation module, although BabelNet seems very useful in order to obtain interconnected language-specific resources, some issues have been identified which will have to be dealt with in the future. First of all, languages with a very productive compositional process (e.g., German) have BabelNet synsets for which there is no direct correspondence in other languages (in other words, they correspond to more than one synset). Second, and partly as a consequence of the first issue, not all BabelNet synsets correspond to a term in a specific language. Third and last, the procedure for compiling BabelNet synsets can be optimized: if a sequence of lexical units is considered as a multiword unit, then synsets are duplicated (one synset is assigned for each single unit and another one for the multiword unit).

As far as analysis is concerned, we plan to incorporate alternative surface-syntactic parsers based on recurrent neural networks [12, 4], which have been found to be particularly beneficiary for out-of-vocabulary words.

Acknowledgements

This work has been supported by the European Commission under the contract number FP7-ICT-610411.

[7] https://catalog.ldc.upenn.edu/LDC2003T05
[8] http://www.multisensorproject.eu/

REFERENCES

[1] M. Ballesteros, B. Bohnet, S. Mille, and L. Wanner, ‘Deep-syntactic parsing’, in *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, Dublin, Ireland, (2014).

[2] M. Ballesteros, B. Bohnet, S. Mille, and L. Wanner, ‘Data-driven deep-syntactic dependency parsing’, *Natural Language Engineering*, 1–36, (2015).

[3] M. Ballesteros, B. Bohnet, S. Mille, and L. Wanner, ‘Data-driven sentence generation with non-isomorphic trees’, in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 387–397, Denver, Colorado, (May–June 2015). Association for Computational Linguistics.

[4] M. Ballesteros, C. Dyer, and N.A. Smith, ‘Improved transition-based parsing by modeling characters instead of words with lstms’, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 349–359, Lisbon, Portugal, (September 2015). Association for Computational Linguistics.

[5] B. Bohnet, A. Björkelund, J. Kuhn, W. Seeker, and S. Zarriess, ‘Generating non-projective word order in statistical linearization’, in *Proceedings of the 2012 Joint Conference on EMNLP and CoNLL*, pp. 928–939, Jeju Island, Korea, (July 2012). Association for Computational Linguistics.

[6] B. Bohnet and J. Nivre, ‘A transition-based system for joint Part-of-Speech tagging and labeled non-projective dependency parsing’, in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 1455–1465, Jeju Island, Korea, (2012).

[7] N. Bouayad-Agha, G. Casamayor, and L. Wanner, ‘Content Determination from an Ontology-based Knowledge Base for the Generation of Football Summaries’, in *Proceedings of the 13th European Natural Language Generation Workshop (ENLG)*, pp. 27–81, Stroudsburg, PA, USA, (2011). Association for Computational LInguistics.

[8] J.C.K. Cheung and G. Penn, ‘Unsupervised sentence enhancement for automatic summarization’, in *Proceedings of the 2014 Conference on EMNLP*, pp. 775–786, Doha, Qatar, (October 2014). Association for Computational Linguistics.

[9] D. Dannélls, ‘The value of weights in automatically generated text structures’, *Lecture Notes in Computer Science*, **5449 LNCS**, 233–244, (2009).

[10] S. Demir, S. Carberry, and K.F. McCoy, ‘A Discourse-aware Graph-based Content Selection Framework’, in *Proceedings of the 6th International Natural Language Generation Conference*, pp. 17–27, Stroudsburg, PA, USA, (2010). Association for Computational LInguistics.

[11] Michelangelo Diligenti, Marco Gori, and Marco Maggini, ‘A unified probabilistic framework for web page scoring systems’, *IEEE Transactions on knowledge and data engineering*, **16**(1), 4–16, (2004).

[12] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N.A. Smith, ‘Transition-based dependency parsing with stack long short-term memory’, in *Proceedings of the 53rd Annual ACL Meeting and the 7th International Joint Conference on Natural Language Processing*, pp. 334–343, Beijing, China, (July 2015). Association for Computational Linguistics.

[13] Charles J. Fillmore, Collin F. Baker, and Hiroaki Sato, ‘The FrameNet database and software tools’, in *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, pp. 1157–1160, Las Palmas, Canary Islands, Spain, (2002).

[14] A. Freitas, J.G. Oliveira, E. Curry, S. O’Riain, and J.C. Pereira da Silva, ‘Treo: Combining entity-search, spreading activation and semantic relatedness for querying linked data’, in *In: 1st Workshop on Question Answering over Linked Data (QALD-1) Workshop at 8th Extended Semantic Web Conference (ESWC*, (2011).

[15] K. Ganesan, C. Zhai, and J. Han, ‘Opinosis: A graph based approach to abstractive summarization of highly redundant opinions’, in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 340–348, Beijing, China, (August 2010). Coling 2010 Organizing Committee.

[16] P.-E. Genest and G. Lapalme, ‘Framework for abstractive summarization using text-to-text generation’, in *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pp. 64–73. Association for Computational Linguistics, (2011).

[17] P.-E. Genest and G. Lapalme, ‘Fully abstractive approach to guided summarization’, in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 354–358, Jeju Island, Korea, (July 2012). Association for Computational Linguistics.

[18] B. J Grosz, S. Weinstein, and A. K Joshi, ‘Centering: A framework for modeling the local coherence of discourse’, *Computational linguistics*, **21**(2), 203–225, (1995).

[19] V. Gupta and G.S. Lehal, ‘A survey of text summarization extractive techniques’, *Journal of Emerging Technologies in Web Intelligence*, **2**(3), 258–268, (2010).

[20] Atif Khan, Naomie Salim, and Yogan Jaya Kumar, ‘A framework for multi-document abstractive summarization based on semantic role labelling’, *Applied Soft Computing*, **30**, 737–747, (2015).

[21] P. Kingsbury and M. Palmer, ‘From TreeBank to PropBank’, in *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, pp. 1989–1993, Las Palmas, Canary Islands, Spain, (2002).

[22] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N.A. Smith, ‘Toward abstractive summarization using semantic representations’, in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1077–1086, Denver, Colorado, (May–June 2015). Association for Computational Linguistics.

[23] Fei Liu and Yang Liu, ‘From extractive to abstractive meeting summaries: Can it be done by sentence compression?’, in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 261–264, Suntec, Singapore, (August 2009). Association for Computational Linguistics.

[24] Simon M., A. Burga, and L. Wanner, ‘AnCora-UPF: A multi-level annotation of Spanish’, in *Proceedings of the 2nd International Conference on Dependency Linguistics (DepLing)*, pp. 217–226, Prague, Czech Republic, (2013).

[25] Kathleen R. McKeown, ‘The text system for natural language generation: An overview’, in *Proceedings of the 20th Annual Meeting on Association for Computational Linguistics*, ACL ’82, pp. 113–120, Stroudsburg, PA, USA, (1982). Association for Computational Linguistics.

[26] Y. Mehdad, G. Carenini, and R.T. Ng, ‘Abstractive summarization of spoken and written conversations based on phrasal queries’, in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1220–1230, Baltimore, Maryland, (June 2014). Association for Computational Linguistics.

[27] Igor Mel’čuk, *Dependency Syntax: Theory and Practice*, State University of New York Press, Albany, 1988.

[28] Simon Mille and Leo Wanner, ‘Towards large-coverage detailed lexical resources for data-to-text generation’, in *Proceedings of the First International Workshop on Data-to-text Generation*, Edinburgh, Scotland, (2015).

[29] A. Moro, A.l Raganato, and R. Navigli, ‘Entity linking meets word sense disambiguation: a unified approach’, *Transactions of the Association for Computational Linguistics*, **2**, 231–244, (2014).

[30] Roberto Navigli and Simone Paolo Ponzetto, ‘Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network’, *Artificial Intelligence*, **193**, 217–250, (2012).

[31] M. O’Donnell, C. Mellish, J. Oberlander, and A. Knott, ‘ILEX: an Architecture for a Dynamic Hypertext Generation System’, *Natural Language Engineering*, **7**, 225–250, (2001).

[32] Martha Palmer, ‘Semlink: Linking Propbank, VerbNet and FrameNet’, in *Proceedings of the Generative Lexicon Conference (GenLex-09)*, Pisa, Italy, (2009).

[33] Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, and Janet Hitzeman, ‘Centering: A Parametric Theory and Its Instantiations’, *Computational Linguistics*, **30**(3), 309–363, (2004).

[34] Ehud Reiter and Robert Dale, *Building Natural Language Generation Systems*, Cambridge University Press, New York, NY, USA, 2000.

[35] H. Saggion and G. Lapalme, ‘Generating indicative-informative summaries with sumum’, *Computational linguistics*, **28**(4), 497–526, (2002).

[36] Karin Kipper Schuler, *VerbNet: A broad-coverage, comprehensive verb lexicon*, Ph.D. dissertation, University of Pennsylvania, 2005.

[37] Lucy Vanderwende, Michele Banko, and Arul Menezes, ‘Event-centric summary generation’, *Working notes of DUC*, 127–132, (2004).

[38] L. Wanner, B. Bohnet, N. Bouayad-Agha, F. Lareau, and D. Nicklaß, ‘MARQUIS: Generation of user-tailored multilingual air quality bulletins’, *Applied Artificial Intelligence*, **24**(10), 914–952, (2010).