Framework for Abstractive Summarization using Text-to-Text Generation

Pierre-Etienne Genest, Guy Lapalme
RALI-DIRO
Université de Montréal
P.O. Box 6128, Succ. Centre-Ville
Montréal, Québec
Canada, H3C 3J7
{genestpe,lapalme}@iro.umontreal.ca

Abstract

We propose a new, ambitious framework for abstractive summarization, which aims at selecting the content of a summary not from sentences, but from an abstract representation of the source documents. This abstract representation relies on the concept of Information Items (INIT), which we define as the smallest element of coherent information in a text or a sentence. Our framework differs from previous abstractive summarization models in requiring a semantic analysis of the text. We present a first attempt made at developing a system from this framework, along with evaluation results for it from TAC 2010. We also present related work, both from within and outside of the automatic summarization domain.

1 Introduction

Summarization approaches can generally be categorized as extractive or abstractive (Mani, 2001). Most systems developped for the main international conference on text summarization, the Text Analysis Conference (TAC) (Owczarzak and Dang, 2010), predominantly use sentence extraction, including all the top-ranked systems, which make only minor post-editing of extracted sentences (Conroy et al., 2010) (Gillick et al., 2009) (Genest et al., 2008) (Chen et al., 2008).

Abstractive methods require a deeper analysis of the text and the ability to generate new sentences, which provide an obvious advantage in improving the focus of a summary, reducing its redundancy and keeping a good compression rate. According to a recent study (Genest et al., 2009b), there is an empirical limit intrinsic to pure extraction, as compared to abstraction. For these reasons, as well as for the technical and theoretical challenges involved, we were motivated to come up with an abstractive summarization model.

Recent abstractive approaches, such as sentence compression (Knight and Marcu, 2000) (Cohn and Lapata, 2009) and sentence fusion (Barzilay and McKeown, 2005) or revision (Tanaka et al., 2009) have focused on rewriting techniques, without consideration for a complete model which would include a transition to an abstract representation for content selection. We believe that a "fully abstractive" approach requires a separate process for the analysis of the text that serves as an intermediate step before the generation of sentences. This way, content selection can be applied to an abstract representation rather than to original sentences or generated sentences.

We propose the concept of *Information Items* (INIT) to help define the abstract representation. **An INIT is the smallest element of coherent information in a text or a sentence**. It can be something as simple as some entity's property or as complex as a whole description of an event or action. We believe that such a representation could eventually allow for directly answering queries or guided topic aspects, by generating sentences targeted to address specific information needs.

Figure 1 compares the workflow of our approach with other possibilities. Extractive summarization consists of selecting sentences directly from the

Workshop on Monolingual Text-To-Text Generation, pages 64–73,

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 64–73,
Portland, Oregon, 24 June 2011. ©2011 Association for Computational Linguistics

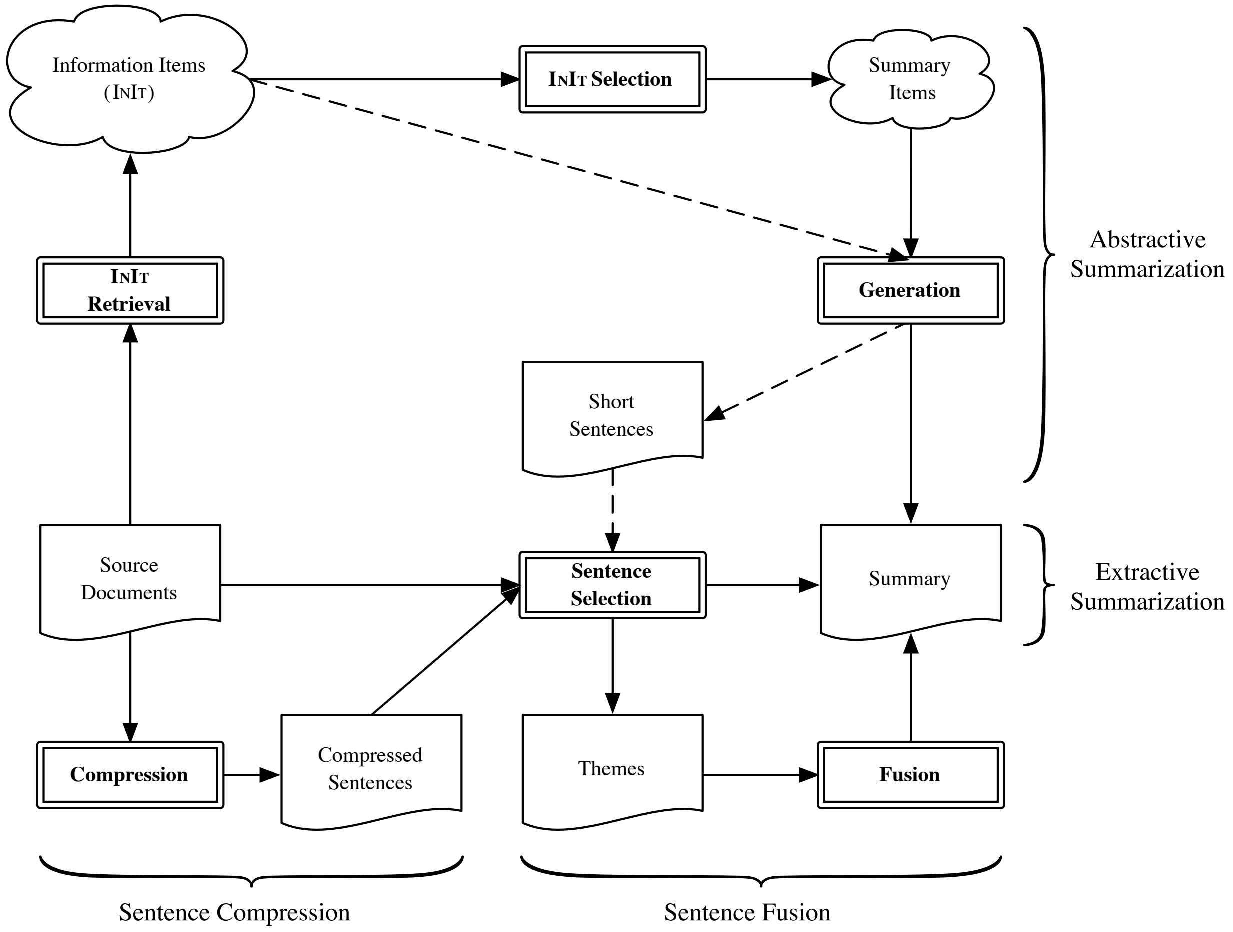

Figure 1: Workflow diagram of our suggested approach for abstractive summarization, compared to pure extractive summarization, sentence compression, and sentence fusion for summarization. The dashed line represents the simplified framework used in our first attempt at abstractive summarization (see section 2.4).

source documents and generating a summary from them. Sentence compression first compresses the sentences and chooses from those and the source documents' sentences to form a summary; it may also be completed in the reverse order, which is to select sentences from the source documents and then compress them for the summary. Sentence fusion first identifies themes (clusters of similar sentences) from the source documents and selects which themes are important for the summary (a process similar to the sentence selection of centroid-based extractive summarization methods (Radev et al., 2004)) and then generates a representative sentence for each theme by sentence fusion.

Our proposed abstractive summarization approach is fundamentally different because the selection of content is on Information Items rather than on sentences. The text-to-text generation aspect is also changed. Instead of purely going from whole sentences to generated sentences directly, there is now a text planning phase that occurs at the conceptual level, like in Natural Language Generation (NLG).

This approach has the advantage of generating typically short, information-focused sentences to produce a coherent, information rich, and less redundant summary. However, the difficulties are great: it is difficult for a machine to properly extract information from sentences at an abstract level, and text generated from noisy data will often be flawed. Generating sentences that do not all sound similar and generic is an additional challenge that we have for now circumvented by re-using the original sentence structure to a large extent, which is a type of text-to-text generation. Even considering those difficulties, we believe that efforts in abstractive summarization constitute the future of summarization research, and thus that it is worthwhile to work towards that end.

In this paper, we present our new abstractive summarization framework in section 2. Section 3 describes and analyses our first attempt at using this framework, for the TAC 2010 multi-document news summarization task, followed by the competition's results in section 4. In this first attempt, we simplified the framework of section 2 to obtain early results which can help us as we move forward in this project. Related work is discussed in section 5, and we conclude in section 6.

2 Abstractive Summarization Framework

Our proposed framework for fully abstractive summarization is illustrated in figure 1. This section discusses how each step could be accomplished.

2.1 InIt Retrieval

An Information Item is the smallest element of coherent information in a text or a sentence. This intentionally vague definition leaves the implementation details to be decided based on resources available. The goal is to identify all entities in the text, their properties, predicates between them, and characteristics of the predicates. This seemingly unreachable goal, equivalent to machine reading, can be limited to the extent that we only need InIts to be precise and accurate enough to generate a summary from them.

The implementation of InIts is critical, as everything will depend on the abstract information available. Semantic Role Labeling (SRL) and predicate-logic analysis of text are two potential candidates for developing InIt Retrieval. Word-sense disambiguation, co-reference resolution and an analysis of word similarity seem important as well to complement the semantic analysis of the text.

2.2 InIt Selection

Given an analysis of the source documents that leads to a list of InIts, we may now proceed to select content for the summary. Frequency-based models, such as those used for extractive summarization, could be applied to InIt selection instead of sentence selection. This would result in favoring the most frequently occurring entities, predicates, and properties.

InIt selection could also easily be applied to tasks such as query-driven or guided summarization, in which the user information need is known and the summarization system attempts to address it. With smaller building blocks (InIts rather than sentences), it would be much easier to tailor summaries so that they include only relevant information.

2.3 Generation

Planning, summary planning in our case, provides the structure of the generated text. Most InIts do not lead to full sentences, and need to be combined into a sentence structure before being realized as text. Global decisions of the InIt selection step now lead to local decisions as to how to present the information to the reader, and in what order.

Text generation patterns can be used, based on some knowledge about the topic or the information needs of the user. One could use heuristic rules with different priority levels or pre-generated summary scenarios, to help decide how to structure sentences and order the summary. We believe that machine learning could be used to learn good summary structures as well.

Once the detailed planning is completed, the summary is realized with coherent syntax and punctuation. This phase may involve text-to-text generation, since the source documents' sentences provide a good starting point to generate sentences with varied and complex structures. The work of (Barzilay and McKeown, 2005) on sentence fusion shows an example of re-using the same syntactical structure of a source sentence to create a new one with a slightly different meaning.

2.4 First Attempt at Abstractive Summarization

The three-step plan that we laid down is very hard, and instead of tackling it head on, we decided to focus on certain aspects of it for now. We followed a simplified version of our framework, illustrated by the dashed line in Figure 1. It defers the content selection step to the selection of generated short sentences, rather than actually doing it abstractly as

Original Sentence The Cypriot airliner that crashed in Greece may have suffered a sudden loss of cabin pressure at high altitude, causing temperatures and oxygen levels to plummet and leaving everyone aboard suffocating and freezing to death, experts said Monday.

Information Items

1. airliner – crash – *null* (Greece, August 15, 2005)
2. airliner – suffer – loss (Greece, August 15, 2005)
3. loss – cause – *null* (Greece, August 15, 2005)
4. loss – leave – *null* (Greece, August 15, 2005)

Generated Sentences

1. A Cypriot airliner crashed.
2. A Cypriot airliner may have suffered a sudden loss of cabin pressure at high altitude.
3. A sudden loss of cabin pressure at high altitude caused temperatures and oxygen levels to plummet.
4. A sudden loss of cabin pressure at high altitude left everyone aboard suffocating and freezing to death.

Selected Generated Sentence as it appears in the summary

1. On August 15, 2005, a Cypriot airliner crashed in Greece.

Original Sentence At least 25 bears died in the greater Yellowstone area last year, including eight breeding-age females killed by people.

Information Items

1. bear – die – *null* (greater Yellowstone area, last year)
2. person – kill – female (greater Yellowstone area, last year)

Generated Sentences

1. 25 bears died.
2. Some people killed eight breeding-age females.

Selected Generated Sentence as it appears in the summary

1. Last year, 25 bears died in greater Yellowstone area.

Figure 2: Two example sentences and their processing by our 2010 system. In the summary, the date and location associated with an INIT are added to its generated sentence.

planned. The summary planning has to occur after generation and selection, in a *Summary Generation* step not shown explicitly on the workflow.

We have restricted our implementation of INITs to dated and located subject–verb–object(SVO) triples, thus relying purely on syntactical knowledge, rather than including the semantics required for our framework. Dates and locations receive a special treatment because we were interested in news summarization for this first attempt, and news articles are factual and give a lot of importance to date and location.

We did not try to combine more than one INIT in the same sentence, relying instead on short, to-the-point sentences, with one INIT each. Figure 2 shows two examples of sentences that were generated from a source document sentence using the simplified abstractive summarization framework.

At first glance, the simplified version of our approach for generating sentences may seem similar to sentence compression. However, it differs in three important ways from the definition of the task of compression usually cited (Knight and Marcu, 2000):

- Our generated sentences intend to cover only one item of information and not all the important information of the original sentence.
- An input sentence may have several generated sentences associated to it, one for each of its INITS, where it normally has only one compressed sentence.
- Generated sentences sometimes include words that do not appear in the original sentence (like 'some' in the second example), whereas sentence compression is usually limited to word deletion.

3 Abstractive Summarization at TAC 2010

Our first attempt at full abstractive summarization took place in the context of the TAC 2010 multi-document news summarization task. This section describes briefly each module of our system, while (Genest and Lapalme, 2010) provides the implementation details.

3.1 INIT Retrieval

An INIT is defined as a dated and located subject–verb–object triple, relying mostly on syntactical analyses from the MINIPAR parser (Lin, 1998) and linguistic annotations from the GATE information extraction engine (Cunningham et al., 2002).

Every verb encountered forms the basis of a candidate INIT. The verb's subject and object are extracted, if they exist, from the parse tree. Each INIT is also tagged with a date and a location, if appropriate.

Many candidate INITS are rejected, for various reasons: the difficulty of generating a grammatical and meaningful sentence from them, the observed unreliability of parses that include them, or because it would lead to incorrect INITS most of the time. The rejection rules were created manually and cover a number of syntactical situations. Cases in which bad sentences can be generated remain, of course, even though about half the candidates are rejected. Examples of rejected Inits include those with verbs in infinitive form and those that are part of a conditional clause. Discarding a lot of available information is a significant limitation of this first attempt, which we will address as the first priority in the future.

3.2 Generation

From each INIT retrieved, we directly generate a new sentence, instead of first selecting INITS and planning the summary. This is accomplished using the original parse tree of the sentence from which the INIT is taken, and the NLG realizer SimpleNLG (Gatt and Reiter, 2009) to generate an actual sentence. Sample generated sentences are illustrated in Figure 2.

This process – a type of text-to-text generation – can be described as translating the parts that we want to keep from the dependency tree provided by the parser, into a format that the realizer understands. This way we keep track of what words play what role in the generated sentence and we select directly which parts of a sentence appear in a generated sentence for the summary. All of this is driven by the previous identification of INITS. We do not include any words identified as a date or a location in the sentence generation process, they will be generated if needed at the summary generation step, section 3.4.

Sentence generation follows the following steps:

- Generate a Noun Phrase (NP) to represent the subject if present
- Generate a NP to represent the object if present
- Generate a NP to represent the indirect object if present
- Generate a complement for the verb if one is present and only if there was no object
- Generate the Verb Phrase (VP) and link all the components together, ignoring anything else present in the original sentence

NP Generation

Noun phrase generation is based on the subtree of its head word in the dependency parse tree. The head in the subtree becomes the head of the NP and children in its parse subtree are added based on manual rules that determine which children are realized and how.

Verb Complement Generation

When an INIT has no object, then we attempt to find another complement instead, in case the verb would have no interesting meaning without a complement. The first verb modifier that follows it in the sentence order is used, including for example prepositional phrases and infinitive clauses.

VP Generation

Finally, the verb phrases are generated from each verb and some of its children. The NPs generated for the subject, object and indirect object are added, as well as the verb complement if it was generated. If there is an object but no subject, the VP is set to passive, otherwise the active form is always used. The tense (past or present) of the VP is set to the tense of the verb in the original sentence, and most modifiers like auxiliaries and negation are conserved.

3.3 Sentence Selection

To determine which of the generated sentences should be used in the summary, we would have liked to choose from among the INITs directly. For example, selecting the most frequent INIT, or INITs containing the most frequent subject-verb pair seem reasonable at first. However, during development, no such naive implementation of selecting INITs provided satisfactory results, because of the low frequency of those constructs, and the difficulty to compare them semantically in our current level of abstraction. Thus this critical content selection step occurs after the sentence generation process. Only the generated sentences are considered for the sentence selection process; original sentences from the source documents are ignored.

We compute a score based on the frequencies of the terms in the sentences generated from the INITs and select sentences that way. Document frequency (DF) – the number of documents that include an entity in its original text – of the lemmas included in the generated sentence is the main scoring criterion. This criterion is commonly used for summaries of groups of similar documents. The generated sentences are ranked based on their average DF (the sum of the DF of all the unique lemmas in the sentence, divided by the total number of words in the sentence). Lemmas in a stop list and lemmas that are included in a sentence already selected in the summary have their DF reduced to 0, to avoid favoring frequent empty words, and to diminish redundancy in the summary.

3.4 Summary Generation

A final summary generation step is required in this first attempt, to account for the planning stage and to incorporate dates and locations for the generated sentences.

Sentence selection provides a ranking of the generated sentences and a number of sentences intentionally in excess of the size limit of the summary is first selected. Those sentences are ordered by the date of their INIT when it can be determined. Otherwise, the day before the date of publication of the article that included the INIT is used instead. All generated sentences with the same known date are grouped in a single coordinated sentence. The date is included directly as a pre-modifier “On *date*,” at the beginning of the coordination.

Each INIT with a known location has its generated sentence appended with a post-modifier “in *location*”, except if that location has already been mentioned in a previous INIT of the summary.

At the end of this process, the size of the summary is always above the size limit. We remove the least relevant generated sentence and restart the summary generation process. We keep taking away the least relevant generated sentence in a greedy way, until the length of the summary is under the size limit. This naive solution to never exceed the limit was chosen because we originally believed that our INITs always lead to short generated sentences. However, it turns out that some of the generated summaries are a bit too short because some sentences that were removed last were quite long.

4 Results and Discussion

Here, we present and discuss the results obtained by our system in the TAC 2010 summarization system evaluation. We only show results for the evaluation of standard multi-document summaries; there was also an update task, but we did not develop a specific module for it. After ranking at or near the top with extractive approaches in past years (Genest et al., 2008) (Genest et al., 2009a), we expected a large drop in our evaluation results with our first attempt at abstractive summarization. In general, they are indeed on the low side, but mostly with regards to linguistic quality.

As shown in Table 1, the linguistic quality of our summaries was very low, in the bottom 5 of 43 participating automatic systems. This low linguistic score is understandable, because this was our first try at text generation and abstractive summarization, whereas the other systems that year used sentence extraction, with at most minor modifications made to the extracted sentences.

The cause of this low score is mostly our method for text generation, which still needs to be refined in several ways. The way we identify INITs, as we have already discussed, is not yet developped fully. Even in the context of the methodology outlined in section 3, and specifically 3.2, many improvements can still be made. Errors specific to the current state of our approach came from two major sources: incorrect parses, and insufficiently detailed and sometimes inappropriate rules for "translating" a part of a parse into generated text. A better parser would be helpful here and we will try other alternatives for dependency parsing in future work.

	Pyr.	Ling. Q.	Overall R.
AS	0.315	2.174	2.304
Avg	0.309	2.820	2.576
Best	0.425	3.457	3.174
Models	0.785	4.910	4.760
AS Rank	29	39	29

Table 1: Scores of pyramid, linguistic quality and overall responsiveness for our Abstractive Summarization (AS) system, the average of automatic systems (Avg), the best score of any automatic system (Best), and the average of the human-written models (Models). The rank is computed from amongst the 43 automatic summarization systems that participated in TAC 2010.

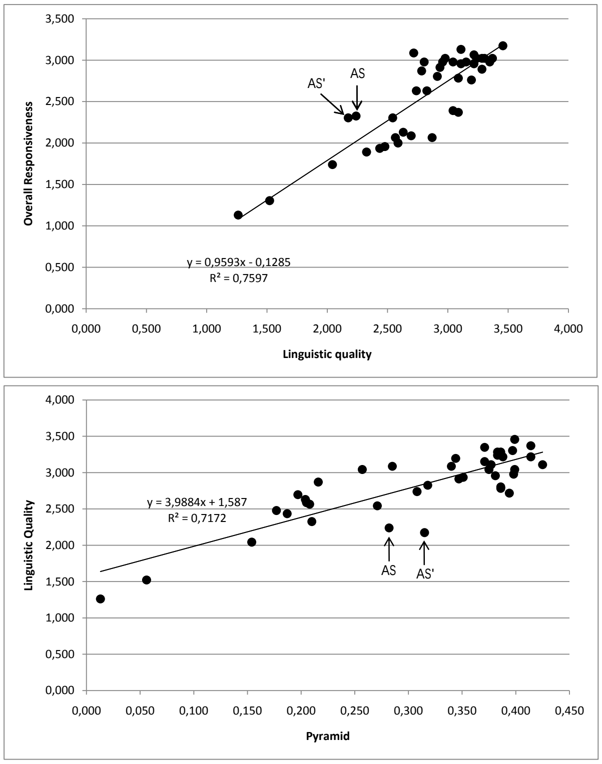

Figure 3: Scatter plots of overall responsiveness with respect to linguistic quality (top) and pyramid score with respect to linguistic quality (bottom), for all the systems competing in TAC 2010. The two runs identified with an arrow, AS and AS', were two similar versions of our abstractive summarization approach.

Although the linguistic quality was very low, our approach was given relatively good Pyramid (Nenkova et al., 2007) (a content metric) and overall responsiveness scores, near the average of automatic systems. This indicates that, even in a rough first try where content selection was not the main focus, our method is capable of producing summaries with reasonably good content and of reasonably good overall quality. There is a correlation between linguistic quality and the other two manual scores for most runs, but, as we can see in Figure 3, the two runs that we submitted stand out, even though linguistic quality plays a large role in establishing the overall responsiveness scores. We believe this to be representative of the great difference of our approach compared to extraction. By extension, following the trend, we hope that increasing the linguistic quality of our approach to the level of the top systems would yield content and overall scores above their current ones.

The type of summaries that our approach produces might also explain why it receives good content and overall scores, even with poor linguistic quality. The generated sentences tend to be short, and although some few may have bad grammar or even little meaning, the fact that we can pack a lot of them shows that INITs give a lot more flexibility to the content selection module than whole sentences, that only few can fit in a small size limit such as 100 words. Large improvements are to be expected, since this system was developped over only a few months, and we haven't implemented the full scale of our framework described in section 2.

5 Related Work

We have already discussed alternative approaches to abstractive summarization in the introduction. This section focuses on other work dealing with the techniques we used.

Subject–Verb–Object (SVO) extraction is not new. Previous work by (Rusu et al., 2007) deals specifically with what the authors call triplet extraction, which is the same as SVO extraction. They have tried a variety of parsers, including MINIPAR, and they build parse trees to extract SVOs similarly to us. They applied this technique to extractive summarization in (Rusu et al., 2009) by building what the authors call semantic graphs, derived from triplets, and then using said graphs to identify the most interesting sentences for the summary. This purpose is not the same as ours, and triplet extraction was conducted quite superficially (and thus included a lot of noise), whereas we used several rules to clean up the SVOs that would serve as INITs.

Rewriting sentences one idea at a time, as we have done in this work, is also related to the field of text simplification. Text simplification has been associated with techniques that deal not only with helping readers with reading disabilities, but also to help NLP systems (Chandrasekar et al., 1996). The work of (Beigman Klebanov et al., 2004) simplifies sentences by using MINIPAR parses as a starting point, in a process similar to ours, for the purpose of helping information-seeking applications in their own task. (Vickrey and Koller, 2008) applies similar techniques, using a sequence of rule-based simplifications of sentences, to preprocess documents for Semantic Role Labeling. (Siddharthan et al., 2004) uses shallow techniques for syntactical simplification of text by removing relative clauses and appositives, before running a sentence clustering algorithm for multi-document summarization.

The kind of text-to-text generation involved in our work is related to approaches in paraphrasing (Androutsopoulos and Malakasiotis, 2010). Paraphrase generation produces sentences with similar meanings, but paraphrase extraction from texts requires a certain level of analysis. In our case, we are interested both in reformulating specific aspects of a sentence, but also in identifying parts of sentences (INITs) with similar meanings, for content selection. We believe that there will be more and more similarities between our work and the field of paraphrasing as we improve on our model and techniques.

6 Conclusion

We have proposed an ambitious new way of looking at abstractive summarization, with our proposed framework. We believe that this framework aims at the real goals of automatic summarization – controlling the content and structure of the summary. This requires both an ability to correctly analyze text, and an ability to generate text. We have described a first attempt at fully abstractive summarization that relies on text-to-text generation.

We find the early results of TAC 2010 quite satisfactory. Receiving a low linguistic quality score was expected, and we are satisfied with average performance in content and in overall responsiveness. It means that our text-to-text generation was good enough to produce understandable summaries.

Our next step will be to go deeper into the analysis of sentences. Generating sentences should rely less on the original sentence structure and more on the information meant to be transmitted. Thus, we want to move away from the current way we generate sentences, which is too similar to rule-based sentence compression. At the core of moving toward full abstraction, we need to redefine INITs so that they can be manipulated (compared, grouped, realized as sentences, etc.) more effectively. We intend to use tools and techniques that will enable us to find words and phrases of similar meanings, and to allow the generation of a sentence that is an aggregate of information found in several source sentences. In this way, we would be moving away from purely syntactical analysis and toward the use of semantics.

References

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *J. Artif. Int. Res.*, 38:135–187, May.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Beata Beigman Klebanov, Kevin Knight, and Daniel Marcu. 2004. Text simplification for information-seeking applications. In Robert Meersman and Zahir Tari, editors, *Proceedings of Ontologies, Dabases, and Applications of Semantics (ODBASE) International Conference*, volume 3290 of *Lecture Notes in Computer Science*, pages 735–747, Agia Napa, Cyprus, October. Springer.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics - Volume 2*, COLING '96, pages 1041–1044, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shouyuan Chen, Yuanming Yu, Chong Long, Feng Jin, Lijing Qin, Minlie Huang, and Xiaoyan Zhu. 2008. Tsinghua University at the Summarization Track of TAC 2008. In *Proceedings of the First Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *J. Artif. Int. Res.*, 34(1):637–674.

John M. Conroy, Judith D. Schlesinger, Peter A. Rankel, and Dianne P. O'Leary. 2010. CLASSY 2010: Summarization and metrics. In *Proceedings of the Third Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, USA.

Albert Gatt and Ehud Reiter. 2009. SimpleNLG: a Realisation Engine for Practical Applications. In *ENLG '09: Proceedings of the 12th European Workshop on Natural Language Generation*, pages 90–93, Morristown, NJ, USA. Association for Computational Linguistics.

Pierre-Etienne Genest and Guy Lapalme. 2010. Text generation for abstractive summarization. In *Proceedings of the Third Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

Pierre-Etienne Genest, Guy Lapalme, Luka Nerima, and Eric Wehrli. 2008. A Symbolic Summarizer for the Update Task of TAC 2008. In *Proceedings of the First Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

Pierre-Etienne Genest, Guy Lapalme, Luka Nerima, and Eric Wehrli. 2009a. A symbolic summarizer with 2 steps of sentence selection for TAC 2009. In *Proceedings of the Second Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

Pierre-Etienne Genest, Guy Lapalme, and Mehdi Yousfi-Monod. 2009b. HexTac: the Creation of a Manual Extractive Run. In *Proceedings of the Second Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

David Gillick, Benoit Favre, Dilek-Hakkani Tür, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/UTD Summarization System at TAC 2009. In *Proceedings of the Second Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 703–710. AAAI Press.

Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Proc. Workshop on the Evaluation of Parsing Systems*, Granada.

Inderjeet Mani. 2001. *Automatic Summarization*, volume 3 of *Natural Language Processing*. John Benjamins Publishing Company.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4, May.

Karolina Owczarzak and Hoa Trang Dang. 2010. Overview of the TAC 2009 summarization track. In *Proceedings of the Third Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology. http://www.nist.gov/tac/publications/.

Dragomir R. Radev, Hongyan Jing, Malgorzata Stys, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938.

Delia Rusu, Lorand Dali, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. 2007. Triplet extraction from sentences. *Proceedings of the 10th International Multiconference "Information Society – IS 2007"*, A:218–222, October.

Delia Rusu, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. 2009. Semantic graphs derived from triplets with application in document summarization. *Informatica*, 33, October.

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hideki Tanaka, Akinori Kinoshita, Takeshi Kobayakawa, Tadashi Kumano, and Naoto Kato. 2009. Syntax-driven sentence revision for broadcast news summarization. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation*, UCNLG+Sum '09, pages 39–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Vickrey and Daphne Koller. 2008. Sentence Simplification for Semantic Role Labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, Ohio, June. Association for Computational Linguistics.