Fully Abstractive Approach to Guided Summarization

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

This paper shows that full abstraction can be accomplished in the context of guided summarization. We describe a work in progress that relies on Information Extraction, statistical content selection and Natural Language Generation. Early results already demonstrate the effectiveness of the approach.

1 Introduction

In the last decade, automatic text summarization has been dominated by extractive approaches that rely purely on shallow statistics. In the latest evaluation campaign of the Text Analysis Conference[1] (TAC), the top systems were considered only "barely acceptable" by human assessment (Owczarzak and Dang, 2011). The field is also getting saturated near what appears to be a ceiling in performance. Systems that claim to be very different from one another have all become statistically indistinguishable in evaluation results. An experiment (Genest et al., 2009) found a performance ceiling to pure sentence extraction that is very low compared to regular (abstractive) human summaries, but not that much better than the current best automatic systems.

Abstractive summarization has been explored to some extent in recent years: sentence compression (Knight and Marcu, 2000) (Cohn and Lapata, 2009), sentence fusion (Barzilay and McKeown, 2005) or revision (Tanaka et al., 2009), and a generation-based approach that could be called sentence splitting (Genest and Lapalme, 2011). They are all rewriting techniques based on syntactical analysis, offering little improvement over extractive methods in the content selection process.

We believe that a fully abstractive approach with a separate process for the analysis of the text, the content selection, and the generation of the summary has the most potential for generating summaries at a level comparable to human. For the foreseeable future, we think that such a process for full abstraction is impossible in the general case, since it is almost equivalent to perfect text understanding. In specific domains, however, an approximation of full abstraction is possible.

This paper shows that full abstraction can be accomplished in the context of guided summarization. We propose a methodology that relies on Information Extraction and Natural Language Generation, and discuss our early results.

2 Guided Summarization

The stated goal of the guided summarization task at TAC is to motivate a move towards abstractive approaches. It is an oriented multidocument summarization task in which a category is attributed to a cluster of 10 source documents to be summarized in 100 words or less. There are five categories: Accidents and Natural Disasters, Attacks, Health and Safety, Endangered Resources, and Investigations/Trials. Each category is associated with a list of aspects to address in the summary. Figure 1 shows the aspects for the Attacks category. We use this specification of categories and aspects to accomplish domain-specific summarization.

[1] www.nist.gov/tac

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 354–358,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

2.1 WHAT: what happened
2.2 WHEN: date, time, other temporal placement markers
2.3 WHERE: physical location
2.4 PERPETRATORS: individuals or groups responsible for the attack
2.5 WHY: reasons for the attack
2.6 WHO_AFFECTED: casualties (death, injury), or individuals otherwise negatively affected
2.7 DAMAGES: damages caused by the attack
2.8 COUNTERMEASURES: countermeasures, rescue efforts, prevention efforts, other reactions

Figure 1: Aspects for TAC's guided summarization task, category 2: Attacks

3 Fully Abstractive Approach

Guided summarization categories and aspects define an information need, and using Information Extraction (IE) seems appropriate to address it. The idea to use an IE system for summarization can be traced back to the FRUMP system (DeJong, 1982), which generates brief summaries about various kinds of stories; (White et al., 2001) also wrote abstractive summaries using the output of an IE system applied to events such as natural disasters. In both cases, the end result is a generated summary from the information available. A lot of other work has instead used IE to improve the performance of extraction-based systems, like (Barzilay and Lee, 2004) and (Ji et al., 2010).

What is common to all these approaches is that the IE system is designed for a specific purpose, separate from summarization. However, to properly address each aspect requires a system designed specifically for that task. To our knowledge, tailoring IE to the needs of abstractive summarization has not been done before. Our methodology uses a rule-based, custom-designed IE module, integrated with Content Selection and Generation in order to write short, well-written abstractive summaries.

Before tackling these, we perform some preprocessing on the cluster of documents. It includes: cleaning up and normalization of the input using regular expressions, sentence segmentation, tokenization and lemmatization using GATE (Cunningham et al., 2002), syntactical parsing and dependency parsing (collapsed) using the Stanford Parser (de Marneffe et al., 2006), and Named Entity Recognition using Stanford NER (Finkel et al., 2005). We have also developed a date resolution engine that focuses on days of the week and relative terms.

3.1 Information Extraction

Our architecture is based on *Abstraction Schemes*. An abstraction scheme consists of IE rules, content selection heuristics and one or more generation patterns, all created by hand. Each abstraction scheme is designed to address a theme or subcategory. Thus, rules that extract information for the same aspect within the same scheme will share a similar meaning. An abstraction scheme aims to answer one or more aspects of its category, and more than one scheme can be linked to the same aspect.

Figure 2 shows two of the schemes that we have created. For the scheme **killing**, the IE rules would match X as the perpetrator and Y as a victim for all of the following phrases: X `killed` Y, Y `was assassinated by` X, and `the murder of` X `by` Y. Other schemes have similar structure and purpose, such as **wounding**, **abducting**, **damaging** and **destroying**. To create extraction rules for a scheme, we must find several verbs and nouns sharing a similar meaning and identify the syntactical position of the roles we are interested in. Three resources have helped us in designing extraction rules: a thesaurus to find semantically related nouns and verbs; VerbNet (Kipper et al., 2006), which provides amongst other things the semantic roles of the syntactical dependents of verbs; and a hand-crafted list of aspect-relevant word stems provided by the team that made CLASSY (Conroy et al., 2010).

Schemes and their extraction rules can also be quite different from this first example, as shown with the scheme **event**. This scheme gathers the basic information about the attack event: WHAT category of attack, WHEN and WHERE it occurred. A list of *key words* is used to identify words that imply an attack event, while a list of EVENT NOUNs is used to identify specifically words that refer to a type of attack.

Scheme: killing

Information Extraction	SUBJ(`kill`, X)	→	WHO(X)
	OBJ(`kill`, Y)	→	WHO_AFFECTED(Y)
	SUBJ(`assassinate`, X)	→	WHO(X)
	OBJ(`assassinate`, Y)	→	WHO_AFFECTED(Y)
		⋮	
	PREP_OF(`murder`, Y)	→	WHO_AFFECTED(Y)
	PREP_BY(`murder`, X)	→	WHO(X)
		⋮	
Content Selection	Select best candidates for *kill verb*, WHO(X) and WHO_AFFECTED(Y)		
Generation	X *kill verb* Y		

Scheme: event

Information Extraction	PREP_IN(*key word*, X), LOCATION(X)	→	WHERE(X)
	PREP_IN(*key word*, X), ORGANIZATION(X)	→	WHERE(X)
	PREP_AT(*key word*, X), LOCATION(X)	→	WHERE(X)
	PREP_AT(*key word*, X), ORGANIZATION(X)	→	WHERE(X)
	DEP(*key word*, Y), DATE(Y)	→	WHEN(Y)
	EVENT NOUN(Z)	→	WHAT(Z)
Content Selection	Select best candidates for `at` or `in`, WHERE(X), WHEN(Y) and WHAT(Z)		
Generation	`On` Y, Z `occurred at/in` X		

Figure 2: Abstraction schemes **killing** and **event**. The information extraction rules translate preprocessing annotations into candidate answers for a specific aspect. Content selection determines which candidate will be included in the generated sentence for each aspect. Finally, a pattern is used to determine the structure of the generated sentence. Notation: `word or lemma`, variable, *group of words*, PREDICATE OR ASPECT. Note that the predicate DEP matches any syntactical dependency and that *key words* refer to a premade list of category-relevant verbs and nouns.

3.2 Content Selection

A large number of candidates are found by the IE rules for each aspect. The content selection module selects the best ones and sends them to the generation module. The basic heuristic is to select the candidate most often mentioned for an aspect, and similarly for the choice of a preposition or a verb for generation. More than one candidate may be selected for the aspect WHO_AFFECTED, the victims of the attack. Several heuristics are used to avoid redundancies and uninformative answers.

News articles may contain references to more than one event of a given category, but our summaries describe only one. To avoid mixing candidates from two different event instances that might appear in the same cluster of documents, we rely on dates. The ancestors of a date in the dependency tree are associated with that date, and excluded from the summary if the main event occurs on a different date.

3.3 Generation

The text of a summary must be fluid and feel natural, while being straightforward and concise. From our observation of human-written summaries, it also does not require a great deal of originality to be considered excellent by human standards. Thus, we have designed straightforward generation patterns for each scheme. They are implemented using the SimpleNLG realizer (Gatt and Reiter, 2009), which takes a sentence structure and words in their root form as input and gives a sentence with resolved agreements and sentence markers as output. The greatest difficulty in the structure is in realizing noun phrases. The content selection module selects a lemma that should serve as noun phrase head, and its number, modifiers and specifier must be determined during generation. Frequencies and heuristics are again used to identify appropriate modifiers, this time from all those used with that head within the source documents. We apply the constraint that the

On April 20, 1999, a massacre occurred at Columbine High School.
Two student gunmen killed 12 students, a teacher and themselves.

On November 2, 2004, a brutal murder occurred in Amsterdam.
A gunman stabbed and shot Dutch filmmaker Theo van Gogh.
A policeman and the suspect were wounded.

On February 14, 2005, a suicide car bombing occurred in Beirut.
Former Lebanese Prime Minister Rafik Hariri and 14 others were killed.

Figure 3: Brief fully abstractive summaries on clusters D1001A-A, D1039G-A and D1043H-A, respectively on the Columbine massacre, the murder of Theo van Gogh and the assassination of Rafik Hariri.

combination of number and modifiers chosen must appear at least once as an IE rule match.

As for any generated text, a good summary also requires a text plan (Hovy, 1988) (McKeown, 1985). Ours consists of an ordering of the schemes. For example, an Attack summary begins with the scheme **event**. This ordering also determines which scheme to favor in the case of redundancy, e.g. given that a building was both damaged and destroyed, only the fact that is was destroyed will be mentioned.

4 Results and Discussion

We have implemented this fully abstractive summarization methodology. The abstraction schemes and text plan for the Attack category are written in an XML document, designed to easily allow the addition of more schemes and the design of new categories. The language processing of the source documents and the domain-specific knowledge are completely separate in the program.

Our system, which is meant as a proof of concept, can generate useful summaries for the Attack category, as can be seen in Figure 3. The key elements of information are present in each case, stated in a way that is easy to understand.

These short summaries have a high density of information, in terms of how much content from the source documents they cover for a given number of words. For example, using the most widely used content metric, Pyramid (Nenkova et al., 2007), the two sentences generated for the cluster D1001A-A contain 8 Semantic Content Units (SCU) for a weighted total of 30 out of a maximum of 56, for a raw Pyramid score of 0.54. Only 3 of the 43 automatic summaries beat this score on this cluster that year (the average was 0.31). Note that the summaries that we compare against contain up to 100 words, whereas ours is only 21 words long. We conclude that our method has the potential for creating summaries with much greater information density than the current state of the art.

In fact, our approach does not only have the potential to increase a summary's coverage, but also its linguistic quality and the reader satisfaction as well, since the most relevant information now appears at the beginning of the summary.

5 Conclusion and Future Work

We have developed and implemented a fully abstractive summarization methodology in the context of guided summarization. The higher density of information in our short summaries is one key to address the performance ceiling of extractive summarization methods. Although fully abstractive summarization is a daunting challenge, our work shows the feasibility and usefulness of this new direction for summarization research.

We are now expanding the variety and complexity of the abstraction schemes and generation patterns to deal with more aspects and other categories. We should then be able to compare on a greater scale the output of our system with the ones produced by other automatic systems and by humans on all the clusters used at TAC 2010 and 2011.

6 Acknowledgements

The authors want to thank Dr. Eduard Hovy, of ISI, and Prof. Kathy McKeown, of Columbia University, for fruitful discussions on abstractive summarization, and Dr. Judith Schlesinger and Dr. John Conroy, both of the IDA / Center for Computing Sciences, for providing us with their hand-crafted list of category- and aspect-relevant keywords.

References

R. Barzilay and L. Lee. 2004. Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization. *eprint arXiv:cs/0405039*, May.

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

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.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology*. The Stanford Natural Language Processing Group.

Gerald DeJong, 1982. *An Overview of the FRUMP System*, pages 149–176. Lawrence Erlbaum.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

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. 2011. Framework for Abstractive Summarization using Text-to-Text Generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 64–73, Portland, Oregon, USA, June. Association for Computational Linguistics.

Pierre-Etienne Genest, Guy Lapalme, and Mehdi Yousfi-Monod. 2009. 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.

Eduard H. Hovy. 1988. Planning coherent multisentential text. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, pages 163–169, Morristown, NJ, USA. Association for Computational Linguistics.

Heng Ji, Juan Liu, Benoit Favre, Dan Gillick, and Dilek Hakkani-Tur. 2010. Re-ranking summaries based on cross-document information extraction. In Pu-Jen Cheng, Min-Yen Kan, Wai Lam, and Preslav Nakov, editors, *Information Retrieval Technology*, volume 6458 of *Lecture Notes in Computer Science*, pages 432–442. Springer Berlin / Heidelberg. 10.1007/978-3-642-17187-1_42.

Karen Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending VerbNet with Novel Verb Classes. In *LREC 2006*.

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.

Kathleen R. McKeown. 1985. Discourse strategies for generating natural-language text. *Artif. Intell.*, 27:1–41, September.

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. 2011. Overview of the TAC 2011 summarization track: Guided task and aesop task. In *Proceedings of the Fourth Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology. http://www.nist.gov/tac/publications/.

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.

Michael White, Tanya Korelsky, Claire Cardie, Vincent Ng, David Pierce, and Kiri Wagstaff. 2001. Multi-document summarization via information extraction. In *Proceedings of the first international conference on Human language technology research*, HLT '01, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.