

Task Knowledge in Abstractive 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 discusses the path towards abstractive summarization and proposes a new knowledge-based methodology called K-BABS as a step forward on this path. We propose to use both world knowledge, to identify useful content, and task knowledge, to filter out unreliable content, to generate more accurate summaries. This approach was implemented for guided summarization. The evaluation shows that, used in combination with a state-of-the-art system, our K-BABS system significantly improves content coverage in the summaries.

1 Introduction

Abstractive summarization is “one of the main challenges to solve” (Lloret and Palomar, 2012) in text summarization research. Its main difficulty is to aim for a semantic understanding of the documents rather than relying on syntactic rewriting techniques.

This paper discusses steps of increasing complexity on the path towards abstractive summarization, to help position various efforts from the past, the present and the future in a consistent manner. We also introduce the **Knowledge-Based Abstractive Summarization (K-BABS)** methodology, which revisits previous work on using Information Extraction for abstractive summarization, by also providing a means for selecting and filtering summary content more robustly, in a multi-document setting.

1.1 The ITG Model for Summarization

(Spärck Jones, 1999) considers that the basic model of an automatic text summarizer has the following

three stages, forming the acronym ITG:

1. *Interpretation* of the source text into a source text representation
2. *Transformation* of the source text representation into a summary representation
3. *Generation* of the summary text from the summary representation

Her model provides a “common means” for comparing systems based on “the real logic underlying [them]”. The I stage involves a mapping from natural language to an abstract representation of the text, be it based on syntax, semantics, or something equivalently relevant. The kind of transformation that is expected in the T stage is a compressive operation that may involve reasoning, inference, generalization, simplification, etc. Finally, the G stage maps back from an abstract representation of the summary into natural language.

2 The Path Towards Abstractive Summarization

Abstractive summarization has been used to refer to a range of different approaches, from performing edits of source document sentences, to generating new structure and surface forms. For this reason, the more specific terms extractive, semi-extractive and non-extractive are used instead in this discussion.

2.1 Extractive Summarization

In the last two decades, text summarization has been dominated by extractive approaches that select a few sentences in the original source. Graphs, centroids, machine learning, linear programming and

other statistics-based algorithms have become the state of the art. They tend to favor a bag-of-words or bigram representation of the text and make use of rudimentary knowledge resources. Recent surveys (Nenkova and McKeown, 2011) (Lloret and Palomar, 2012) offer a more detailed overview.

Extractive summarization cannot hope to reach human-level performance. (Genest et al., 2009) measured the performance of human-selected extracts in the news domain, showing them to be unsatisfactory, which highlights the inherent limit of extractive approaches. With regards to the ITG model, the interpretation of the source text in extractive summarization does not lead to a representation other than lexical, and the transformation step as defined above is reduced to a selection process involving no truly abstract representation of the content.

2.2 Semi-Extractive Summarization

Semi-extractive summarization techniques also aim to construct the summary directly from extracted text, though they do not limit the selection to whole sentences. Several approaches have been described to rearrange sentences, such as phrase deletion (Pollock and Zamora, 1975), sentence compression (Cohn and Lapata, 2009), sentence splitting (Genest and Lapalme, 2011), sentence fusion (Barzilay and McKeown, 2005) (Filippova, 2010), and sentence revision (Tanaka et al., 2009).

These approaches rely on a syntactic manipulation of the source text to form new sentences. They aim to fit more salient information in the summary by removing extraneous details and by merging similar information from different sources, thus also avoiding redundancies that are typical of extractive summarization. From the standpoint of the ITG model, the operations performed can hardly be considered a conceptual transformation from a source text representation to a summary representation, because they are conducted at a purely syntactic level.

2.3 Non-Extractive Summarization

Non-extractive summarization encompasses a large space of approaches that has been only scarcely explored as of yet, mostly because of the technical challenge it represents. These approaches all have in common an interpretation stage that involves some form of text understanding, and a generation stage

that has the potential for producing linguistic structures and vocabulary that have not been observed in the source documents. Once again, the transformation stage may be sometimes quite elementary, in the less complex approaches. Having access to some form of conceptual representation of the source text and/or of the summary, it is often possible, in non-extractive summarization, to organize the structure of the summary in a meaningful way, as opposed to (semi-)extractive approaches, which often find sentence ordering more difficult. However, non-extractive methods face an even higher risk of generating unintelligible or false summaries.

Some non-extractive summarization systems have focused on adding indicative information about the source documents (Kan et al., 2001) (Saggion and Lapalme, 2002), as opposed to informative content. (Zhang et al., 2013) proposed to summarize Twitter discussions by classifying related speech acts, and reporting separately the most salient statements, questions, suggestions and comments made by the tweet authors, in order to generate short summaries that are both informative and indicative.

More recently, (Mehdad et al., 2014) generated indicative summaries of conversations using a word-graph based approach. While showing interesting results in the conversation domain, the generation method used would probably lead to ungrammatical sentences too often to be competitive against extractive approaches in a domain with typically well-constructed sentences, and where the summaries are purely informative, such as news.

FRUMP (DeJong, 1982), SUMMONS (Radev and McKeown, 1998) and RIPTIDES (White et al., 2001) are informative, non-extractive summarization systems that rely on an Information Extraction (IE) system (not specifically designed for the purpose of summarization). The summarization task was tackled from the angle of natural language generation from data, showing that they have a true interpretation step within the ITG model. FRUMP performs some minor transformation in which events can be inferred, and some selection and comparison of information is performed in SUMMONS and RIPTIDES in the context of a multi-document task, but this accounts only for a fraction of the work; in all cases, the text and summary representations are the same. Moreover, what *can* be extracted by

Information Extraction is considered as being what *should* appear in the summary. FRUMP and SUMMONS have not been formally evaluated against other automatic summarization systems. RIPTIDES showed improvement over a simple extractive baseline.

2.4 Future of Non-Extractive Summarization

The path described so far is only the beginning, and much remains to be done in many areas.

First, the interpretation stage needs to be better. A full machine understanding of the source documents is not required for reaching near human-level performance, but syntactic parsing and shallow semantic annotations are perhaps too limiting. Semantic role labeling has improved, but its accuracy remains too low for direct use in extracting content from documents. Accurate semantic parsing would allow more direct access to the meaningful content of the source.

Second, the transformation stage needs a radical improvement. World knowledge and knowledge about the summarization task itself (what to keep, what to filter), are essential elements, and this knowledge needs to be passed along to the summarizer somehow. Task knowledge is where we are trying to make improvements with K-BABS.

Finally, the generation stage itself could in principle be improved, in order to handle the synthesis into natural language of an unlexicalized abstract representation of meaning or content. Without a description of such a representation, it is hard to predict precisely what will be needed.

3 Knowledge-Based Abstractive Summarization (K-BABS)

We have designed a new abstractive summarization methodology that takes direct control over the transformation stage of the ITG model, a methodology that makes use of both world knowledge and task knowledge. This section describes a system that implements the K-BABS methodology.

3.1 Task: Guided Summarization

The Text Analysis Conference¹ (TAC) 2011's guided summarization task provides a good setting for beginning work on abstractive summarization

¹www.nist.gov/tac

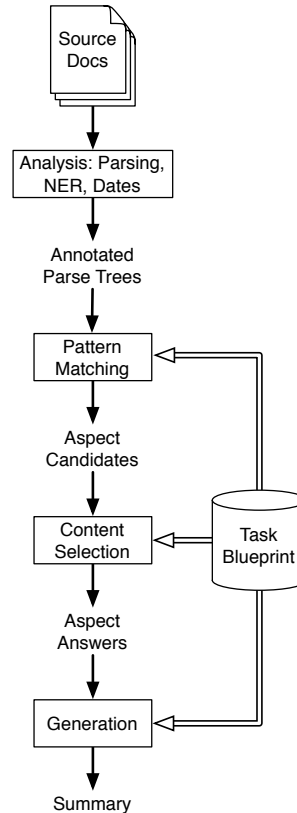


Figure 1: Workflow for K-BABS's architecture.

because of its category-specific aspects. Guided summarization is an oriented multidocument task in which a category is attributed to each cluster of 10 source documents to summarize in 100 words or less. The K-BABS system covers two of the five categories of the task: Accidents and Natural Disasters, and Attacks. For each category, a list of *aspects* to cover in the summary is given; e.g. for ATTACKS: What, When, Where, Perpetrators, Why, Who was affected, Damages, and Countermeasures.

3.2 Architecture

The K-BABS architecture is split into three modules, as seen in Figure 1: Analysis (interpretation stage), which provides an intermediate representation of the source documents; Task Blueprint, which provides world knowledge; and the summarization system itself, which provides task knowledge, by performing pattern matching, content selection and generation.

3.3 Analysis

The analysis module performs the interpretation stage of the ITG model. For our implementation, we rely on the tools freely available today. The main component is Dependency Parsing, using the “collapsed dependencies” of the Stanford Parser (de Marneffe et al., 2006). The dependency parses provide the (somewhat) abstract representation on which pattern matching is later performed in the transformation stage. Lemmatization using the GATE morphological analysis tool (Cunningham et al., 2002) and Named entity resolution of “person”, “location”, “organization” and “date”, using Stanford NER (Finkel et al., 2005) are also performed.

3.4 Task Blueprint and Abstraction Schemes

The task blueprint is a data base for providing world knowledge about each category and each aspect of the guided summarization task. It implicitly determines the content that is deemed salient to appear in the summary of a given category by leaving out what is not covered. It also defines lists of lemmas and multiple syntactic structures that define a semantic equivalence, contextualized by the known category (domain) of the source documents.

This knowledge is encoded in *abstraction schemes*, which define a set of pattern matching rules where the LHS is a pattern in the source text representation (in our case: dependency parses), and the RHS is an aspect with a variable that gets instantiated. For each abstraction scheme, a *generation template* describes the structure of a sentence that can be generated from the facts extracted during pattern matching. For example (Figure 3), one of our abstraction schemes separately looks for the best description of a type of attack, the most salient location, and the most salient event-related date, and combines them to generate one very compact sentence based on this template: “On <date>, an <attacktype> occurred in <location>”.

The task blueprint also includes a generation plan for each category, such as the one in Figure 3, which defines the structure of the summary by listing the schemes in the order in which their generated sentence will appear in the summary.

Two task blueprints were manually created in our first implementation of K-BABS, for the ATTACKS

and ACCIDENTS AND NATURAL DISASTERS categories. The TAC 2010 clusters were used as a development set and the TAC 2011 clusters were reserved (and never observed throughout) as a test set. The process mostly involved determining what patterns to use and especially trying them on the development set and other documents of the same domain to check whether it leads to the desired content. Populating lists of words, either lemma groups or stoplists, was another significant part of the manual effort, whereas coming up with generation templates was straightforward. About a dozen abstraction schemes were written for each category, each scheme with about 4–6 pattern matching rules, and a total of about 25 lemma lists totalling over 200 lemmas per category. As we discuss in section 3.6, we believe that this process should be replaced with (semi-)automatic pattern discovery.

3.5 Summarization System

The summarization system’s input consists of the analysis of the source documents, and the task blueprint prepared for its category, which provides the world knowledge necessary to write an abstractive summary. The top-level algorithm for writing a summary is to read the generation plan and sequentially generate a sentence for each of the abstraction schemes, when possible and when it would not be redundant to do so. Applying pattern matching rules and generation templates from the abstraction schemes is relatively straightforward, but doing only that would lead to bad summaries.

How redundancy is handled is at the heart of summarization. K-BABS is designed so that information about similar entities and events can be grouped and generated into a single sentence. This is exactly the kind of advantage that abstractive summarization has over extractive methods, and it is provided by the task blueprint.

The role of the summarization system in reducing redundancy is related to the extra step from Figure 1 called Content Selection, that performs the kind of filtering needed for outputting more useful information than noise (falsehoods and unintelligible sentences). The design and calibration of the content selection step, which can be seen as providing task knowledge, is a novelty of our approach.

During pattern matching, an abstraction scheme’s

Scheme: killing	
Pattern Matching	SUBJ_RELATIONS(<i>kill_verbs</i> , X) → WHO(X)
	OBJ_RELATIONS(<i>kill_verbs</i> , Y) → WHO_AFFECTED(Y)
	PREP_OF(<i>murder_nouns</i> , Y) → WHO_AFFECTED(Y)
	PREP_BY(<i>murder_nouns</i> , X) → WHO(X)
Generation Template	X <i>kill_verbs</i> Y
Scheme: event	
Pattern Matching	PREP_IN(<i>event_lemmas</i> , X), LOCATION(X) → WHERE(X)
	PREP_IN(<i>event_lemmas</i> , X), ORGANIZATION(X) → WHERE(X)
	PREP_AT(<i>event_lemmas</i> , X), LOCATION(X) → WHERE(X)
	PREP_AT(<i>event_lemmas</i> , X), ORGANIZATION(X) → WHERE(X)
	DEP(<i>event_lemmas</i> , Y), DATE(Y) → WHEN(Y)
	EVENT NOUN(Z) → WHAT(Z)
Generation Template	On Y, Z occur at/in X

Figure 2: Abstraction schemes **killing** and **event**. The pattern matching rules define how to detect aspect candidates from the dependency parsing annotations and semantic information detected by the analysis module. The generation template defines how to realize a sentence for output. Notation: word or lemma, variable, lemma group, PREDICATE OR ASPECT. The special predicate DEP is the set of all the syntactic relations from the parser, and the lemma group *event_lemmas* is a set of many verbs and nouns strongly related to the category.

Abstraction Scheme	Example structure of the generated sentence
event	On a date, an attack/murder/shooting/etc. occurred at/in a location.
beingHitGeneric	X was attacked/hit/struck.
killing	X killed/murdered/shot/etc. Y.
dying	X died.
injuring	X wounded/injured Y.
destroying	X was damaged/destroyed.
arresting	X was arrested.
suspecting	X was suspected of conducting the attack.
responsibilityClaim	X claimed responsibility for the attack.
helping	X sent/provided/offered support/aid/help.
beingRescued	X was rescued.
evacuating	X was evacuated.

Figure 3: An ordered list of abstraction schemes that serves as the generation plan for the category ATTACKS.

rules are applied over all the source documents to provide a list of candidates for each aspect that the abstraction scheme intends to cover (see Figure 2).

Content selection is the process of selecting from the aspect candidates (usually heads of noun phrases), the one that will become the *aspect answer* for each aspect, such as which instantiation of the variable X in WHO(X) to use for generation, and the selected lemma for a lemma group, such as which *kill_verb* to use (stabbing, shooting, bombing, etc.). Depending on the abstraction scheme, it is possible that we look for more than one candidate (e.g. look only for one perpetrator of an attack (possibly a

word referring to a group rather than an individual), but allow for a whole set of victims to be reported by the system), in which case all of them are outputted.

Selecting candidates is based on candidate frequency, lexical overlap, and domain-specific stoplists. The most frequently observed candidate not on a stoplist is selected for generation. Candidates that have not been matched at least twice cannot be selected; this is especially important in filtering out candidates in cases where the aspect answer may include more than one aspect candidate. Head nouns which appear as noun-noun modifiers of other candidates are favored over others. And if a head noun

can be a noun-noun modifier of some other head noun (as observed in the source documents), those two candidates will not be repeated in a list of candidate answers.

Once head nouns are selected, content selection also determines how to build the whole noun phrase corresponding to an aspect answer. Thus, a similar process based on highest frequency is used to select the determiner, count, modifier, etc. of the candidate's noun phrase for generation.

Sentences are generated after all the elements of the abstraction scheme's generation template have been selected and filtered, including typically a verb and noun phrases for the syntactic positions of subject and/or object. When a required element is missing, no sentence is generated. If more than one subject or more than one object has been selected, the noun phrases are coordinated when permitted for that abstraction scheme. Sentences are realized using the SimpleNLG (Gatt and Reiter, 2009) library.

Applying the generation plan also participates in content selection. In the case of the *ATTACKS* category, for example, the generation plan prohibits the same entity to be declared as having both been killed by someone, and also as having died, or as having also been injured by that same person. Similarly, if the summary declares that some Mr. X has shot someone, the generation plan prohibits the abstraction scheme **suspecting** from generating that Mr. X is suspected of conducting the attack.

3.6 Scalability of K-BABS

The K-BABS approach is scalable to the extent that the same architecture can be re-used with (semi-)automatically generated task blueprints, even though there was, in this first version, significant manual effort involved in designing pattern matching rules for each category. For somewhat similar categories, some abstraction schemes can be repeated with only few changes needed. Very different categories, however, require basically starting over from scratch, e.g. *HEALTH&SAFETY* compared to *ATTACKS* and *ACCIDENTS/NATURAL DISASTERS*. Thankfully, there are alternatives to the burden of manually constructing resources.

Automatically discovering domain-specific blueprints for the K-BABS methodology is the next logical step. Recent work on finding event patterns

for abstractive summarization (Pighin et al., 2014) has been very promising, and would provide a good setting for semi-automatically acquiring abstraction scheme patterns. A number of previous works in other fields can also provide a starting point to the unsupervised learning of abstraction schemes, such as automatic schemata acquisition (Mooney and DeJong, 1985), narrative schema acquisition (Chambers and Jurafsky, 2008), information extraction template acquisition (Chambers and Jurafsky, 2011) (Saggion, 2013), and extraction of domain patterns (Filatova et al., 2006).

3.7 Hybrid Summaries

Our K-BABS system generates very short summaries that often lack in coverage. This is expected of a methodology that favors precision over recall in its content selection process, but it makes comparisons with other approaches difficult. To this end, we developed a hybrid approach that uses sentence extraction to lengthen the summary to 100 words. This hybrid summarizer always outputs the summary provided by K-BABS first, and then completes the summary using a state-of-the-art extractive system.

The automatic summarizer with the best overall responsiveness from the participants of TAC 2011 was CLASSY (Conroy et al., 2011). The CLASSY team generously offered to produce and share with us hybrid summaries for the 18 clusters in the data set. In the K-BABS/CLASSY hybrid, the abstractive summary is considered as a bag-of-words for avoiding redundancy in the extractive summarization part. Figure 4 shows two hybrid summaries generated on the test set.

4 Evaluation

The system built based on K-BABS was evaluated following the standard methodology developed in the DUC and TAC evaluation campaigns. This includes the manual scoring of overall responsiveness, linguistic quality and content coverage.

The evaluation set contains 18 clusters of 10 NewsWire articles from TAC 2011, never seen during development. There were 44 TAC clusters that year, but only the ones with the category *ATTACKS* or *ACCIDENTS AND NATURAL DISASTERS* were included, because other categories are not yet cov-

Cluster Topic	Summary generated by the K-BABS/CLASSY hybrid
8. Cyclone Sidr	<i>On November 15, 2007, a cyclone occurred in Bangladesh. The cyclone caused a storm. The powerful cyclone hit the southwestern coast of Bangladesh. Hundreds died. Homes were destroyed. Thousands were evacuated.</i> A fierce cyclone packing extreme winds and torrential rain smashed into Bangladesh's southwestern coast Thursday, wiping out homes and trees in what officials described as the worst storm in years. A powerful cyclone with strong winds started pounding on Bangladesh's south and southwestern coast from Thursday evening. The money will go to German relief organisations working in cooperation with local partners to alleviate suffering caused by Cyclone Sidr.
37. Crane Collapse	<i>On March 15, 2008, a construction accident occurred. A towering crane crashed. Four people died. 24 people were injured. A five-story building and parts of five buildings were demolished. A building was evacuated.</i> A piece of steel fell and sheared off one of the ties holding it to the building, causing the structure to detach and topple, said Stephen Kaplan, an owner of the Reliance Construction Group. Neighborhood residents said they had complained to the city several times about the construction at the site, saying crews worked illegal hours and the building was going up too fast.

Figure 4: Summaries generated by the K-BABS/CLASSY hybrid for 2 clusters of the test set. The *italics* part of each summary is the output from K-BABS, while the rest is generated by CLASSY. Cluster 8 is an example of great overlap between the abstractive part and the extractive part, while cluster 37 shows complementarity.

ered by our system.

The K-BABS system, CLASSY (the best performing automatic system on that task so far), their hybrid, an extractive baseline that outputs the first few sentences of the most recent document, and TAC 2011's human-written model summaries were manually evaluated. The results are shown in Figure 5.

Ten human assessors not involved in the project were asked to score the summaries according to the same guidelines as those used at TAC 2011 for overall responsiveness and linguistic quality. Each summary was evaluated by 3 assessors on this scale: very poor (1), poor, barely acceptable, good, very good (5). For the content score, modified Pyramid (Nenkova et al., 2007) was used, according to the TAC 2011 annotations. To account for the varying summary lengths, content density was also measured, defined as the modified Pyramid of a summary divided by its length in words.

5 Discussion

5.1 Statistical Significance of Results

As suggested in (Rankel et al., 2011), a Wilcoxon Signed Rank test, a paired statistical test for distributions that may not be normal (necessary in our case with only 18 data points), was performed on several

pairs of averages presented in the Figure 5.

From a statistical standpoint, the significant differences are:

1. The abstractive/extractive hybrid summaries have higher content (p -value < 0.05) than the purely extractive summaries of CLASSY.
2. K-BABS has a significantly higher content density than the other automatic approaches.
3. K-BABS may or may not have better linguistic quality than CLASSY, with a p -value of 0.12.
4. Unsurprisingly, the human-written summaries are far ahead in all respects.

The K-BABS system's higher content density, by a factor of almost 2 to 1, is striking. Of course, its summaries were very short, but they also include almost exclusively information that answers a category aspect directly, and little else. This probably explains why putting these sentences at the front in a hybrid setting significantly improves a state-of-the-art system's informative content.

5.2 Qualitative Observations

Figure 4 shows two generated summaries, where the first part is entirely generated by the K-BABS system. The linguistic style these summaries is by

	Overall. R.	Ling. Qual.	Mod. Pyr.	Size	Content Density
K-BABS	2.07	3.67	0.277	22.6	0.0119
CLASSY	3.20	3.39	0.520	98.0	0.0053
K-BABS/CLASSY Hybrid	3.31	3.28	0.600	97.6	0.0061
Extractive baseline	2.70	3.76	0.395	84.5	0.0046
Human-written models	4.54	4.69	-	96.6	-

Figure 5: Scores of overall responsiveness, linguistic quality, modified Pyramid (content), and content density. The size is the average length of the summaries (average number of words).

design telegraphic, generating intentionally short, to-the-point sentences to express content efficiently. However, even given that style, a few summaries remain awkward-sounding or even ungrammatical, which explains why K-BABS does not have a higher linguistic quality score.

What is perhaps a major problem of the abstractive summaries is that some false information is sometimes reported as fact, because of errors that may occur at any point during the processing. This is far less frequent in extractive summaries, where it typically only occurs due to incorrectly resolved coreference between extracted sentences taken from different context. In designing our K-BABS system, we attempted to address this problem by designing very selective content selection rules, but it didn't always succeed on the test data.

The hybrid summaries show how redundancy can still be a major issue, whereas when it is not, it leads to very informative summaries that have all the key points at the beginning in the abstractive part, and additional context and peripheral information are provided by the extracted sentences.

Designing extractive summarizers that aim to provide additional, peripheral information about a topic, given a pre-existing short summary could be an interesting new research direction, that would drive extractive summarization away from relying too heavily on centrality criteria. A recent case study (Cheung and Penn, 2013) suggests that this is precisely what is needed for more robust multi-document summarization.

5.3 Differences between K-BABS and Other Non-Extractive Approaches

One of the key differences with previous works is that the slot fills in K-BABS are open-ended. Many

aspect candidates are considered for inclusion as an aspect answer, allowing for the aggregation of information over varying sources, in combinations that appear in no unique document.

K-BABS makes direct use of redundancy and other criteria to filter aspect candidates, thus partly solving the problem posed by bad parses and incorrect applications of pattern matching rules. The heuristics used for content selection represent a departure from previous work on non-extractive summarization such as SUMMONS (Radev and McKeown, 1998) and RIPTIDES (White et al., 2001), where there was no second guessing of the information extracted.

We believe that it is K-BABS's focus on precision over recall that has led to good performance, even when compared with the state of the art in extractive summarization. No such comparison had been made in previous works.

6 Conclusion

This work has demonstrated both the desirability and the feasibility of abstractive summarization. The K-BABS system's performance, especially in the context of hybrid summarization, is very promising.

Our goal with K-BABS was to focus on improving the transformation stage of the ITG model by implementing more robust content selection. Indeed, our approach includes not only a source of world knowledge (in the task blueprint), but also heuristics that provide task knowledge (in the summarization system). The latter was made possible by having direct access to the information extraction rules used, which is why we designed them manually in this first version. We believe that we took one step forward towards abstractive summarization and automatically discovering rules will be the next one.

References

- Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.
- Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio, June. Association for Computational Linguistics.
- Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 976–986, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jackie Chi Kit Cheung and Gerald Penn. 2013. Towards robust abstractive multi-document summarization: A caseframe analysis of centrality and domain. pages 1233–1242. Association for Computational Linguistics.
- 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, Jeff Kubina, Peter A. Rankel, and Dianne P. O’Leary. 2011. CLASSY 2011 at tac: Guided and multi-lingual summaries and evaluation metrics. In *Proceedings of the Fourth 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.
- Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 207–214, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Katja Filippova. 2010. Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 322–330, Stroudsburg, PA, USA. Association for Computational Linguistics.
- 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 *Proceedings of the 12th European Workshop on Natural Language Generation*, ENLG '09, pages 90–93, Stroudsburg, PA, 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.
- Min-Yen Kan, Kathleen R. McKeown, and Judith L. Klavans. 2001. Applying natural language generation to indicative summarization. In *Proceedings of the 8th European workshop on Natural Language Generation - Volume 8*, EWNLG '01, pages 1–9, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Elena Lloret and Manuel Palomar. 2012. Text summarization in progress: a literature review. *Artif. Intell. Rev.*, 37(1):1–41, January.
- Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. Abstractive summarization of spoken and written conversations based on phrasal queries. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1220–1230. Association for Computational Linguistics.
- Raymond Mooney and Gerald DeJong. 1985. Learning schemata for natural language processing. In *Proceedings of the 9th international joint conference on Artificial intelligence - Volume 1*, IJCAI'85, pages 681–687, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2–3):103–233.

- 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.
- Daniele Pighin, Marco Cornolti, Enrique Alfonseca, and Katja Filippova. 2014. Modelling events through memory-based, open-ie patterns for abstractive summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 892–901. Association for Computational Linguistics.
- J. J. Pollock and A. Zamora. 1975. Automatic abstracting research at chemical abstracts service. *Journal of Chemical Information and Computer Sciences*, 15(4):226–232.
- Dragomir R. Radev and Kathleen R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Comput. Linguist.*, 24(3):470–500.
- Peter Rinkel, John M. Conroy, Eric V. Slud, and Dianne P. O’Leary. 2011. Ranking human and machine summarization systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP ’11, pages 467–473, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Horacio Saggion and Guy Lapalme. 2002. Generating indicative-informative summaries with sumum. *Comput. Linguist.*, 28(4):497–526, December.
- Horacio Saggion. 2013. Unsupervised learning summarization templates from concise summaries. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, USA, June. Association for Computational Linguistics.
- Karen Spärck Jones. 1999. Automatic summarising: Factors and directions. *Advances in Automatic Text Summarization*, pages 1–12.
- 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.
- Renxian Zhang, Wenjie Li, Dehong Gao, and You Ouyang. 2013. Automatic Twitter Topic Summarization With Speech Acts. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(3):649–658.