

- and sub-totals. In Proc. 12th Int. Conf. on Data Engineering, 1996, pp. 152–159.
5. Hurtado C.A., Gutiérrez C., and Mendelzon A. Capturing summarizability with integrity constraints in OLAP. *ACM Trans. Database Syst.*, 30(3):854–886, 2005.
 6. Hurtado C.A. and Mendelzon A.O. Reasoning about summarizability in heterogeneous multidimensional schemas. In Proc. 8th Int. Conf. on Database Theory, 2001, pp. 375–389.
 7. Lenz H-J. and Shoshani A. Summarizability in OLAP and statistical data bases. In Proc. 9th Int. Conf. on Scientific and Statistical Database Management, 1997, pp. 132–143.
 8. Pedersen T.B. and Jensen C.S. Multidimensional data modeling for complex data. In Proc. 15th Int. Conf. on Data Engineering, 336–345, 1999.
 9. Rafanelli M. and Shoshani A. STORM: a statistical object representation model. In Proc. 2nd Int. Conf. on Scientific and Statistical Database Management, 1990, pp. 14–29.
 10. Shoshani A. OLAP and statistical databases: similarities and differences. In Proc. 16th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, 1997, pp. 185–196.

Summarization

JIMMY LIN

University of Maryland, College Park, MD, USA

Synonyms

[Text/document summarization](#); [Automatic abstracting](#); [Distillation](#); [Report writing](#)

Definition

Summarization systems generate condensed outputs that convey important information contained in one or more sources for particular users and tasks. In principle, input sources and system outputs are not limited to text (e.g., keyframe extraction for video summarization), but this entry focuses exclusively on generating *textual* summaries from *textual* sources.

Historical Background

Summarization has a long history dating back to the 1960s, when researchers first started developing computer systems that processed natural language [6,12]. Following a number of decades with comparatively few publications, summarization research entered a new phase in the 1990s. A revival of interest was spurred by the growing availability of text in electronic formats and later the World Wide Web. The enormous quantities of information people come into contact with on a daily basis created a need for applications that help

users cope with the proverbial information overload problem. Summarization systems attempt to address this need.

Foundations

Summarization is a broad and diverse field. Traditionally, it is considered a sub-area of natural language processing, but a significant number of innovations have their origins in information retrieval. This entry is organized as follows: first, various summarization factors are discussed. Next, a tripartite processing model for summarization systems is presented, which provides a basis for discussing general issues. Finally, selected summarization techniques are briefly overviewed.

Summarization Factors

To better understand summarization, it is helpful to enumerate its many dimensions – what Sparck Jones [19] calls “factors”. These factors provide a basis for understanding various automatic methods, and can be grouped into three broad categories: *input*, *purpose*, and *output*. What follows is meant to be an overview of important factors, and not intended to be exhaustive.

Input factors characterize the source of the summaries:

1. *Single versus multiple sources*. For example, one versus multiple reports of the same event.
2. *Genre* (categories of texts) and *Register* (different styles of writing). For example, dissertations versus blogs.
3. *Written versus spoken*. For example, newspaper articles versus broadcast news.
4. *Language*. Sources may be in multiple language.
5. *Metadata*. Sources may be associated with controlled vocabulary keywords, human-assigned category labels.
6. *Structure*. Source structure may be relatively straightforward (e.g., headings and sub-headings) or significantly more complex (e.g., email threads).

Purpose factors characterize the use of summaries (i.e., why they were created):

1. *Indicative versus informative versus evaluative*. Indicative summaries are meant to guide the selection of sources for more in-depth study, whereas informative summaries cover salient information in the sources at some level of detail (and is often meant to replace the original). Evaluative summaries

assess the subject matter of the source and the quality of the work (e.g., a review of a movie).

2. *Generic versus focused.* A generic summary places equal emphasis on different information contained in the sources and provides balanced coverage. Alternatively, a summary might be focused on an information need, i.e., created to answer a question.
3. *Task.* What will the summary be used for? For example, to help write a report or to make a decision.
4. *Audience.* Whom is the summary intended for? For example, experts, schoolchildren, etc.

Output factors characterize system output (note that the input factors are relevant here also, but not repeated):

1. *Extractive versus abstractive.* Extractive summaries consist of text copied from the source material; typically, such approaches are based on shallow analysis. Abstractive summaries contain text that is system-generated, usually based on deeper analysis. Note that these approaches define a continuous spectrum, as many systems employ hybrid methods.
2. *Reduction, coverage, and fidelity.* Reduction, usually measured as a ratio between summary length and source length, is often inversely related to coverage, how much information of interest is preserved in the summary. The summary should also preserve source information accurately.
3. *Coherence.* Does the summary read fluently and grammatically, both syntactically and at the discourse level? For summaries not intended to be fluent prose (e.g., bullets), this factor is less important.

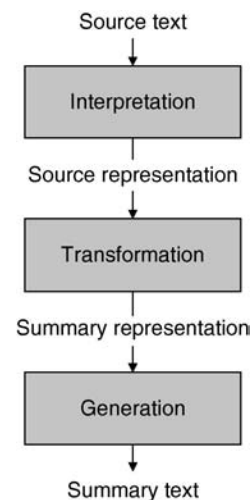
Input, purpose, and output factors together characterize the many dimensions of summarization and provide a basis for subsequent discussions. Note, however, that not all factors figure equally in current summarization systems – for a variety of reasons, the field has focused on some more than others.

Processing Model

Sparck Jones characterizes the process of summarization as a reductive transformation of source text to summary text through content condensation by selection and/or generalization of what is important in

the source [19]. She proposes a tripartite processing model, shown in Figure 1, that serves as a framework for understanding how various summarization techniques fit together (see also [15] for a similar model). Systems first convert source text into the source representation, which is then transformed into the summary representation. Finally, the summary representation is realized as natural language text. Note that these stages do not necessarily map to system components, as the processing model only describes abstract processing tasks. Since this model does not prescribe specific representations or particular processing methods, it is sufficiently general to describe a wide variety of summarization systems while at the same time highlighting important differences.

As previously discussed, input may come from one or multiple sources (the term “documents” is used generically, recognizing that sources may also be speech, email, etc.). Single-document summarization is challenging because simple baselines are often very difficult to improve upon. For example, since news articles are typically written in the “inverse pyramid” style (most important information first), the first sentence or paragraph makes an excellent summary. Frequently, longer documents (e.g., reports) contain “executive summaries”, which nicely capture important information in the documents. Multi-document summarization faces a different set of challenges, the most salient of which is the possibility of redundant information in the sources (e.g., multiple news articles



Summarization. Figure 1. A tripartite processing model for summarization.

about the same event). Frequently, the redundancy is not superficially obvious, but involves paraphrase (different syntactic structures, word choices, etc.). More complex are cases where the information partially overlaps or appears contradictory (e.g., different reports of death tolls). More generally, multi-document summarization requires systems to detect similarities and differences in text.

It is generally assumed that a summarization system is provided the source text. In cases where this is assumption is not met, information retrieval techniques may be used to first select the set of documents to summarize (from a larger collection of documents). However, since most systems assume that input sources are more or less relevant to the task at hand, they may not adequately cope with imperfect retrieval results.

The use of “representation” does not necessarily imply deep linguistic analysis or processing. In fact, most extractive summarization systems adopt a “bag of words” representation at both the source and summary end – that is, text is represented as a vector that has a feature for each word. This representation makes the obviously false assumption that word occurrences are independent and ignores the rich linguistic relationships present in text. Nevertheless, extractive techniques have proven to be effective in various summarization tasks.

With extractive techniques, generation is trivial since systems simply copy material from the source. However, pure extraction often leads to problems in overall coherence of the summary – a frequent issue concerns “dangling” anaphora. Sentences often contain pronouns, which lose their referents when extracted out of context. Worse yet, stitching together decontextualized extracts may lead to misleading interpretation of anaphors (resulting in an inaccurate representation of source information, i.e., low fidelity). Similar issues exist with temporal expressions. Note that these problems become more severe in the multi-document case, since extracts are drawn from different sources. A general approach to addressing these issues involves post-processing extracts, for example, replacing pronouns with their antecedents, replacing relative temporal expression with actual dates, etc. Such techniques, however, can not be considered purely extractive (hence the observation that most systems are, in fact, hybrid).

In general, extractive systems can be characterized as “knowledge-poor”, which is contrasted against “knowledge-rich” approaches. While not synonymous,

abstractive methods tend to be associated with “knowledge-rich” approaches. They involve one or more of the following: detailed linguistic analysis on source text to produce richly annotated structures, incorporation of world knowledge to support the transformation process, or generation of fluent natural language text from abstract representations.

A canonical example of abstractive summarization involves integration with information extraction (IE) systems. Information extraction concerns the automatic identification and creation of template instances from natural language text based on some pre-defined structure. For example, a template for natural disasters might contain “slots” for type, damage, death toll, etc. An IE system would analyze text sources and automatically extract information to fill these templates, in effect, populating a structured database from free text. This process can be viewed as the interpretation stage in the summarization processing model, and the templates themselves can serve as the source representation. A summarization system can then combine information from multiple templates to generate a fluent summary (e.g., [18]).

Abstractive techniques face a number of major challenges, the biggest of which is the representation problem. Systems’ capabilities are constrained by the richness of their representations and their ability to generate such structures – systems cannot summarize what their representations cannot capture. In limited domains, it may be feasible to devise appropriate structures, but a general-purpose solution depends on open-domain semantic analysis. Systems that can truly “understand” natural language are beyond the capabilities of today’s technology.

Finally, coherence of system-generated text is one important output factor in summarization. Coherence is usually taken to mean fluent, grammatically correct prose that “reads well”. This is a tall order, mainly because coherence is very difficult to operationalize. While humans can easily identify incoherent text, they have much more difficulty defining what makes a piece of text coherent. To make matters worse, multiple arrangements of segments might be equally coherent to a human. For extractive techniques, systems must devise an ordering of extracted segments and deal with “out-of-context” issues discussed above. For abstractive techniques, generation of fluent output from an abstract representation is sufficiently difficult that it is considered another sub-area in natural language

processing. Although output coherence is a requirement in both single- and multi-document summarization, the latter presents more problems (particularly for extractive systems) given the variety of sources extracts.

Overview of Selected Techniques

Due to relatively easy access to corpora, most research in summarization over the past two decades has been on written news. As most summarization systems today are primarily extractive, these methods will occupy the bulk of this discussion.

Extractive techniques first segment source text into smaller segments (sentences, paragraphs, etc.), which are then scored according a variety of features, e.g., position in the text [6], term and phrase frequencies [12], lexical chains (degree of lexical-connectedness between various segments) [1], topics present in the text [16], or discourse prominence [14]. A widely adopted approach is to use machine learning techniques to determine the relative importance of various features (the earliest example being [10]).

The features discussed above are relevant for both single- and multi-document summarization, although their relative importance varies with the task. Historically, the summarization field focused on the single-document case first, and then subsequently moved on to multi-document summarization. This move required systems to explicitly model similarities and differences in text to address redundancy, paraphrase, entailment, contradiction, and related linguistic issues. One general approach involves clustering, as exemplified by the MEAD framework [16]. Documents are first clustered to find topics present in the sources. Clusters are represented by their centroids, which are used to rank extracts (along with other features). Maximal Marginal Relevance (MMR) [7] is another effective algorithm, specifically designed for query-focused summaries (i.e., summaries that address an information need). It iteratively selects candidate segments to include in the final summary, balancing relevance and redundancy at each iteration. Redundancy is computed by content similarity between each candidate and the current summary state (using cosine similarity) – thus, candidates containing words already in the summary are penalized. Note that neither MEAD nor MMR explicitly deals with linguistic relationships such as paraphrase, but that issue has been specifically addressed in other work [8].

After scoring and selecting segments from source documents, extractive systems must decide on an ordering in the final system output. Ideally, the output should constitute a coherent piece of text. Simple baselines for ordering segments include extraction order (i.e., by score), temporal order (based on metadata or temporal expressions), and order in source document (preserving source structure). While simple to implement, these techniques frequently yield disfluent summaries. Coherence can be improved by applying computational models of content and discourse [2]. Nevertheless, text structuring is a relatively under-explored area of summarization, particularly due to difficulty in evaluation. As a final note, one possible alternative is to abandon the assumption of summaries as fluent prose, and instead present users with a bulleted list of extracts.

Although open-domain abstractive summarization using deep semantic representations is beyond the current state of the art, a variety of successful abstractive techniques operating on syntactic structures have been developed. Most of these techniques involve parsing source documents and manipulating the resulting parse trees. One popular approach involves “trimming”, or removing inessential structures from the parse tree [9,20] – for example, removing adjunct clauses that do not contribute much information. Other successful techniques include “splicing” fragments from multiple sentences (sometimes across multiple documents) – for example, embedding a simple sentence as a relative clause inside another [3,13]. Of course, these operations are not mutually exclusive. Syntactic manipulations are particularly helpful in multi-document summarization since sentences from different sources might partially overlap, e.g., a sentence contains both redundant and new information. In this case, syntactic operations can potentially deliver the best of both worlds, by eliminating redundant information and preserving new information. However, as Sparck Jones recently noted [19], there has been comparatively little work on abstractive summarization over the last decade.

Additional Readings

Beyond this entry, a number of additional sources are recommended for further reading: slides from a tutorial presentation at SIGIR 2004 [15] provide a good starting point. Special issues of the journal *Information Processing and Management* [19] and *Computational*

Linguistics [17] contain in-depth articles on selected topics. For details on specific summarization techniques, a good place to look is the online proceedings of the Document Understanding Conferences [4], an annual evaluation of summarization systems. A note on references in this entry: since a comprehensive bibliography is impossible due to space limitations, either representative early articles or recent ones are cited (in the latter case, the assumption is that the reader can trace citations backwards).

Key Applications

Summarization technology has a number of applications, many of which are outlined below:

Search result summarization. Search engines typically retrieve thousands of hits (if not more) in response to a user's query. Summarization systems can provide users with an overview of results to support information seeking.

Tools for analytical support. Summarization can be applied to support intelligence analysis, e.g., "prepare a report on recent insurgent activities in Basra", as well as similar activities such as investigative journalism and business intelligence.

Personal information agent. A personal information agent maintains a profile of the user's interest and proactively seeks out information (e.g., retrieving and summarizing relevant news items on a continuous basis).

Accessibility assistance. For example, a visually impaired person might make use of a screen reader augmented with summarization technology for greater efficiency.

Support for handheld devices. Handheld devices such as cell phones and PDAs with small screens could benefit from more condensed information.

Medical applications. Physicians struggle to keep current with the ever-increasing volume of medical literature. Summarization systems can be deployed to assist physicians, e.g., provide an overview of treatment options for a particular disease.

Summarization of meetings. Summarization technology can be coupled with speech recognizers to automatically generate "meeting minutes".

Future Directions

Current research in summarization can be characterized by three broad trends:

Increasing linguistic sophistication. Extractive techniques can benefit from richer features to characterize the appropriateness of a segment for inclusion in the summary – these features come from increasingly detailed linguistic analysis, enabled by advances in language processing technology. Of particular interest is the modeling of linguistic relations such as paraphrase, entailment, and contradiction. Separately, this task has been captured in the PASCAL recognizing textual entailment evaluations.

As discussed above, limitations of extractive methods can be addressed by incorporating abstractive techniques, e.g., manipulation of parse trees. Future developments appear to follow this trend, with increasingly richer representations (enabled by improvements in syntactic, semantic, discourse, and pragmatic analysis). In other words, abstractive summarization will likely be arrived at by successive approximations with hybrid techniques.

Exploration of different genres and domain-specific applications. Recently, researchers have become interested in "informal" text – a broad genre that includes emails, conversational speech, blogs, chat, SMS messages, etc. They are important because an increasing portion of our society's knowledge is captured in these channels. Furthermore, informal text push the frontiers of summarization technology by forcing researchers to develop more general and robust algorithms.

Integration with other language processing components. As technology matures, it becomes feasible to integrate summarization with other components to create more powerful applications. A few examples: integration with speech recognition to summarize TV broadcasts and meetings; integration with machine translation to summarize documents from multiple languages; integration with information retrieval and question answering to produce responses that answer complex questions.

Experimental Results

Summarization is fundamentally experimental in nature, as the effectiveness of different techniques cannot be derived from first principles. Thus, tools for assessing summary quality are critical to ensuring progress, and evaluation methods themselves represent an active area of research.

Methodologies for evaluating system output can be broadly classified into two categories: *intrinsic* and *extrinsic*. In an intrinsic evaluation, system output is

directly evaluated in terms of a set of norms – for example, fluency, coverage of key ideas, or similarity to an “ideal” summary (see [19] for an overview). In particular, the last criteria has been operationalized in ROUGE [11], a commonly used automated metric that compares system output to a number of human-generated “reference” summaries. In contrast, extrinsic evaluations attempt to measure how summarization impacts some other task, for example, helping users determine if a document is relevant (see [5] and references therein). While more informative, extrinsic evaluations are much more difficult to conduct, since it often involves constructing realistic scenarios for summarization systems.

One of the most important driving forces behind summarization research is the existence of annual evaluations that provide a community-wide benchmark to assess progress. Two such evaluations are the Document Understanding Conferences [4] sponsored by the U.S. National Institute of Standards and Technology (NIST), and the NTCIR Project sponsored by Japan’s National Institute of Informatics. Starting in 2008, DUC is replaced by the newly created Text Analysis Conference, also sponsored by NIST.

Data Sets

Instructions for obtaining data from the DUC and NTCIR evaluations can be found on their respective websites.

Cross-references

- ▶ [Information Extraction](#)
- ▶ [Information Retrieval](#)

Recommended Reading

1. Barzilay R. and Elhadad M. Using lexical chains for text summarization. In Proc. ACL/EACL Workshop on Intelligent Scalable Text Summarization, 1997.
2. Barzilay R. and Lee L. Catching the drift: Probabilistic content models, with applications to generation and summarization. In Proc. 2004 Human Language Technology Conf., 2004, pp. 113–120.
3. Barzilay R. and McKeown K.R. Sentence fusion for multidocument news summarization. *Computat. Linguist.*, 31(3):297–327, 2005.
4. Document Understanding Conferences. <http://duc.nist.gov/>.
5. Dorr B.J., Monz C., President S., Schwartz R., and Zajic D. A methodology for extrinsic evaluation of text summarization: Does ROUGE correlate? In Proc. ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, 2005.
6. Edmundson H.P. New methods in automatic extracting. *J. ACM*, 16(2):264–285, 1969.
7. Goldstein J., Mittal V., Carbonell J., and Callan J. Creating and evaluating multi-document sentence extract summaries. In Proc. Int. Conf. on Information and Knowledge Management, 2000, pp. 165–172.
8. Hatzivassiloglou V., Klavans J.L., and Eskin E. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In Proc. Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.
9. Knight K. and Marcu D. Statistics-based summarization – step one: Sentence compression. In Proc. 12th National Conf. on AI, 2000, pp. 703–710.
10. Kupiec J., Pedersen J.O., and Chen F. A trainable document summarizer. In Proc. 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 1995, pp. 68–73.
11. Lin C.Y. and Hovy E. Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proc. 2003 Human Language Technology Conf., 2003, pp. 71–78.
12. Luhn H.P. The automatic creation of literature abstracts. *IBM J. Res. Develop.*, 2(2):159–165, 1958.
13. Mani I., Gates B., and Bloedorn E. Improving summaries by revising them. In Proc. 27th Annual Meeting of the Assoc. for Computational Linguistics, 1999, pp. 558–565.
14. Marcu D. The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts. PhD Thesis, University of Toronto, 1997.
15. Radev D.R. Text summarization. In Tutorial Presentation at the 27th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 2004.
16. Radev D.R., Blair-Goldensohn S., and Zhang Z. Experiments in single and multi-document summarization using MEAD. In Proc. 2001 Document Understanding Conf., 2001.
17. Radev D.R., Hovy E., and McKeown K. Introduction to the special issue on summarization. *Computat. Linguist.*, 28(4):399–408, 2002.
18. Radev D.R. and McKeown K. Generating natural language summaries from multiple on-line sources. *Computat. Linguist.*, 24(3):469–500, 1998.
19. Sparck Jones K. Automatic summarising: The state of the art. *Inf. Process. Manage.*, 43(6):1449–1481, 2007.
20. Zajic D., Dorr B., Lin J., and Schwartz R. Multi-Candidate Reduction: Sentence compression as a tool for document summarization tasks. *Inf. Process. Manage.*, 43(6):1549–1570, 2007.

Summarization Correctness

- ▶ [Summarizability](#)

Summary

- ▶ [Data Sketch/Synopsis](#)