Automated Text Summarization in SUMMARIST

Eduard Hovy and ChinYew Lin
Information Sciences Institute
of the University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292-6695, U.S.A.
tel: +1-310-822-1511
fax: +1-310-823-6714
email: {hovy,cyl}@isi.edu

Abstract

SUMMARIST is an attempt to create a robust automated text summarization system, based on the 'equation': *summarization = topic identification + interpretation + generation*. We describe the system's architecture and provide details of some of its modules.

1 Introduction

1.1 Summary: Extract or Abstract?

The task of a Summarizer is to produce a synopsis of any document (or set of documents) submitted to it. These synopses may range from a list of isolated keywords that indicate the major content of the document(s), through a list of independent single sentences that express the major content, all the way up to a coherent, fully planned and generated paragraph that compresses the document. The more sophisticated a synopsis, the more effort it generally takes to produce.

Several existing systems, including some Web browsers, claim to perform text summarization. However, even a cursory analysis of their output shows that their so-called summaries are actually portions of the text, produced verbatim. While there is nothing wrong with such *extracts*, per se, a truly comprehensive and informative text summary fuses together various concepts of the text into a smaller number of concepts, to form an *abstract*. We define extracts as consisting wholly of portions extracted verbatim from the original (they may be single words or whole passages) and abstracts as consisting of novel phrasings describing the content of the original (which might be paraphrases or fully newly synthesized text). Generally, producing an abstract requires stages of topic fusion and text generation not needed for extracts.

1.2 SUMMARIST

Over the past two years we have been developing the text summarization system SUMMARIST. In this paper, we describe its structure and provide details on the evaluated results of two of its component modules.

The goal of SUMMARIST is to provide both extracts and abstracts for arbitrary English (and later, other-language) input text. SUMMARIST combines symbolic world knowledge (embodied in WordNet, dictionaries, and similar resources) with robust NLP processing (using IR and statistical techniques) to overcome the problems endemic to either approach alone. These problems arise because existing robust NLP methods tend to operate at the word level, and hence miss concept-level generalizations, which are provided by symbolic world knowledge, while on the other hand symbolic knowledge is too difficult to acquire in large enough scale to provide coverage and robustness. For robust summarization, both aspects are needed.

The heart of abstract formation is the interpretation process performed to fuse concepts. This step occurs in the middle of the summarization procedure; to find the appropriate set of concepts in an input text, an initial stage of concept identification and extraction is required; to produce the summary, a final stage of generation is needed. Thus SUMMARIST is based on the following 'equation':

summarization = topic identification + interpretation + generation

This breakdown is motivated as follows:

1. Identification: Select or filter the input to determine the most important, central, topics. For generality we assume that a text can have many (sub)-topics, and that the topic extraction process can be parameterized to include more or fewer of them to produce longer or shorter summaries.

2. Interpretation: Simply aggregating together frequently mentioned portions of the input text does not in itself make an abstract. What are the central, most important, concepts in the following story?

> John and Bill wanted money. They bought ski-masks and guns and stole an old car from a neighbor. Wearing their ski-masks and waving their guns, the two entered the bank, and within minutes left the bank with several bags of $100 bills. They drove away happy, throwing away the ski-masks and guns in a sidewalk trash can. They were never caught.

The popular method of simple word counting would indicate that the story is about ski-masks and guns, both of which are mentioned three times, more than any other word. Clearly, however, the story is about a robbery, and any summary of it must mention this fact. Some process of interpreting the individual words as part of some encompassing concept is required. One such process, word clustering, is an essential technique for topic identification in IR. This technique would match the words "gun", "mask", "money", "caught", "stole", etc., against the set of words that form the so-called signature for the word "robbery". Other, more sophisticated forms of word clustering and fusion are possible, including script matching, deductive inference, and concept clustering.

3. Generation: Two options exist: either the output is a verbatim quotation of some portion(s) of the input, or it must be generated anew. In the former case, no generator is needed, but the output is not likely to be high-quality text (although this might be sufficient for the application).

2 The Structure of SUMMARIST

For each of the three steps of the above 'equation', SUMMARIST uses a mixture of symbolic world knowledge (from WordNet and similar resources) and statistical or IR-based techniques. Each stage employs several different, complementary, methods (SUMMARIST will eventually contain several modules in each stage). To date, we have developed some methods for each stage of processing, and are busy developing additional methods and linking them into a single system. In the next sections we describe one method from each stage. The overall architecture is shown in Figure 1.

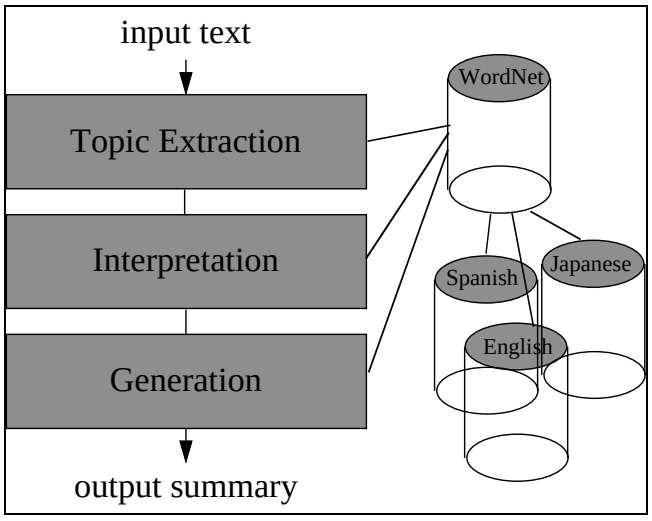

Figure 1. Architecture of SUMMARIST.

2.1 Topic Identification

Several techniques for topic identification have been reported in the literature, including methods based on Position [Luhn 58, Edmundson 69], Cue Phrases [Baxendale 58], word frequency, and Discourse Segmentation [Marcu 97].

We describe here just our work on SUMMARIST's Position module. This method exploits the fact that in some genres, regularities of discourse structure and/or methods of exposition mean that certain sentence positions tend to carry more topic material than others. We defined the *Optimal Position Policy* (OPP) as a list that indicates in what ordinal positions in the text high-topic-bearing sentences occur. We developed a method of automatically training new OPPs, given a collection of genre-related texts with keywords. This work, described in [Lin and Hovy 97a], is the first systematic study and evaluation of the Position method reported.

For the Ziff-Davis corpus (13,000 newspaper articles announcing computer products) we have found that the OPP is

[T1, P2S1, P3S1, P4S1, P1S1, P2S2, {P3S2, P4S2, P5S1, P1S2}, P6S1, ...]

i.e., the title (T1) is the most likely to bear topics, followed by the first sentence of paragraph 2, the first sentence of paragraph 3, etc. In contrast, for the Wall Street Journal the OPP is

[T1, P1S1, P1S2, ...]

Evaluation: We evaluated the OPP method in various ways. In one of them, *coverage* is the fraction of the (human-supplied) keywords that are included verbatim in the sentences selected under the policy. (A random selection policy would extract sentences with a random distribution of topics; a good position policy would extract rich topic-bearing sentences.) We measured the effectiveness of an OPP by taking cumulatively more of its sentences: first just the title, then the title plus P2S1, and so on. In order to determine the effect of multi-word key phrases, we matched using windows of increasing size, from 1 word to 5 words. The resulting coverage scores are shown in Figure 2, broken down by window size. Summing together the multi-word contributions (window sizes 1 to 5) in the top ten sentence positions (R10), the columns reach 95% over an extract of 10 sentences (approx. 15% of a typical Ziff-Davis text): an extremely encouraging result.

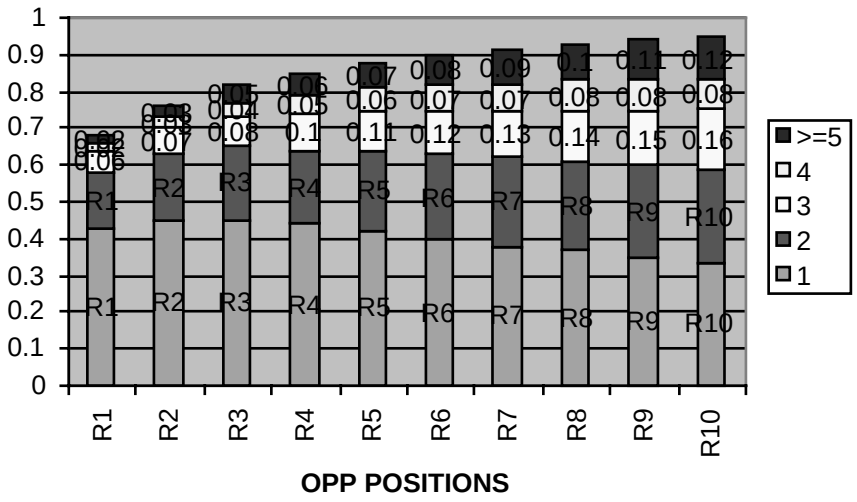

Figure 2. Coverage scores for top ten OPP sentence positions, window sizes 1 to 5.

2.2 Topic Interpretation (Concept Fusion)

The second step in the summarization process is that of concept interpretation. In this step, a collection of extracted concepts are 'fused' into their one (or more) higher-level unifying concept(s). Concept fusion can be as simple as part-whole construction, for example when *wheel*, *chain*, *pedal*, *saddle*, *light*, *frame*, and *handlebars* together fuse to *bicycle*. Generally, though, it is more complex, ranging from direct concept/word clustering as used in IR [Paice 90] to scriptally based inference as in scripts [Schank and Abelson 77].

Fusing topics into one or more characterizing concepts is the most difficult step of automated text summarization. Here, too, a variety of methods can be employed. All of them associate a set of concepts (the *indicators*) with a characteristic generalization (the *fuser* or *head*). The challenge is to develop methods that work reliably and to construct a large enough collection of indicator-fuser sets to achieve effective topic reduction.

SUMMARIST's topic interpretation methods currently include *Concept Wavefront* [Lin 95] and *Concept Signature* [Lin and Hovy 97b].

2.2.1 Concept Counting and the Wavefront

A *topic* is a particular subject that we write about or discuss. To identify the topics of texts, IR researchers make the assumption that the more a word is used in a text, the more important it is in that text. But although word frequency counting operates robustly across different domains without relying on stereotypical text structure or semantic models, they cannot handle synonyms, pronominalization, and other forms of coreferentiality. Furthermore, word counting misses conceptual generalizations:

> *John bought some vegetables, fruit, bread, and milk.→ John bought some groceries.*

The word counting method must be extended to recognize that *vegetables*, *fruit*, etc., relate to *groceries*. Recognizing this inherent problem, people started using Artificial Intelligence techniques [Jacobs 90, Mauldin 91] and statistical techniques [Salton et al. 94] to incorporate semantic relations among words. Following this trend, we have developed a new way to identify topics by counting *concepts* instead of words, and generalizing them using a concept generalization taxonomy. As approximation to such a hierarchy, we employ WordNet [Miller et al. 90] (though we could have used any machine-readable thesaurus) for inter-concept relatedness links. In the limit case, when WordNet does not contain the words, this technique defaults to word counting.

As described in [Lin 95], we locate the most appropriate generalization somewhere in middle of the taxonomy by finding concepts on the *interesting wavefront*, a set of nodes representing concepts that

each generalize a set of approximately equally strongly represented subconcepts (ones that have no obvious dominant subconcept to specialize to).

Evaluation: We selected 26 articles about new computer products from *BusinessWeek* (1993–94) of average 750 words each. For each text we extracted the eight sentences containing the most interesting concepts using the wavefront technique, and comparing them to the contents of a professional's abstracts of these 26 texts from an online service. We developed several weighting and scoring variations and tried various ratio and depth parameter settings for the algorithm. We also implemented a random sentence selection algorithm as a baseline comparison.

The average recall (*R*) and precision (*P*) values over the three scoring variations were *R=0.32* and *P=0.35*, when the system produces extracts of 8 sentences. In comparison, the random selection method had *R=0.18* and *P=0.22* precision in the same experimental setting. While these R and P values are not tremendous, they show that semantic knowledge—even as limited as that in WordNet—does enable improvements over traditional IR word-based techniques. However, the limitations of WordNet are serious drawbacks: there is no domain-specific knowledge, for example to relate *customer*, *waiter*, *cashier*, *food*, and *menu* together with *restaurant*. We thus developed a second technique of concept interpretation, using *category signatures*. We discuss this next.

2.2.2 Interpretation using Signatures

Can one automatically find a set of related words that can collectively be fused into a single word? To test this idea we developed the Concept Signature method [Lin and Hovy 97b]. We defined a signature to be a list of word indicators, each with relative strength of association, jointly associated with the signature head.

To construct signatures automatically, we used a set of 30,000 texts from the *Wall Street Journal* (1987). The Journal editors have classified each text into one of 32 classes— AROspace, BNKing, ENVironment, TELecommunications, etc. We counted the occurrences of each content word (canonicalized morphologically to remove plurals, etc.), in the texts of a class, relative to the number of times they occur in the whole corpus (this is the standard *tf.idf* method). We then selected the top-scoring 300 terms for each category and created a signature with the category name as its head. The top terms of four example signatures are shown in Figure 3. It is quite easy to determine the identity of the signature head just by inspecting the top few signature indicators.

RANK	ARO	BNK	ENV	TEL
1	contract	bank	epa	at&t
2	air_force	thrift	waste	network
3	aircraft	banking	environmental	fcc
4	navy	loan	water	cbs
5	army	mr.	ozone	cable
6	space	deposit	state	bell
7	missile	board	incinerator	long-distance
8	equipment	fslic	agency	telephone
9	mcdonnell	fed	clean	telecomm.
10	northrop	institution	landfill	mci
11	nasa	federal	hazardous	mr.
12	pentagon	fdic	acid_rain	doctrine
13	defense	volcker	standard	service
14	receive	henkel	federal	news
15	boeing	banker	lake	turner

Figure 3. Portions of the signatures of several concepts.

SUMMARIST will use signatures for summary creation as follows. After the topic identification module(s) identify/ies a set of words or concepts, the signature-based concept interpretation module will identify the most pertinent signatures subsuming the topic words, and the signature's head concept will then be used as the summarizing fuser concepts. Matching the identified topic terms against all signature indicators involves several problems, including taking into account the relative frequencies of occurrence and resolving matches with multiple signatures, and specifying thresholds of acceptability.

Evaluation. First, however, we had to evaluate the quality of the signatures formed by our algorithm. Recognizing the similarity of signature recognition to document categorization, we evaluated the effectiveness of each signature by seeing how well it serves as a selection criterion on new texts. As data we used a set of 2,204 previously unseen WSJ news articles from 1988.

For each test text, we created a single-text 'document signature' using the same *tf.idf* measure as before, and then matched this document signature against the category signatures. The closest match provided the class into which the text was categorized. We tested four different matching functions, including a simple *binary* match (count 1 if a term match occurs; 0 otherwise); *curve-fit* match (minimize the difference in occurrence frequency of each term between document and concept signatures), and *cosine* match (minimize the cosine angle in the hyperspace formed when each signature is viewed as a vector and each word frequency specifies the distance along the dimension for that word). These matching functions all provided approximately the same results. The values for Recall and Precision (R=0.756625 and P=0.69309375) are very encouraging and compare well with recent IR results [TREC 95].

Extending this work will require the creation of concept signatures for hundreds, and eventually thousands, of different topics needed for robust summarization. We plan to investigate the effectiveness of a varierty of methods for doing this.

2.3 Summary Generation

The final step in the summarization process is to generate the summary, consisting of the fused concepts, in English. A range of possibilities occurs here, from simple concept printing to sophisticated sentence planning and surface-form realization. Although, as mentioned in Section 1, simple extract summaries require no generation stage, eventually SUMMARIST will contain three generation modules, associated as appropriate with the various levels for various applications:

1. *Topic output:* Sometimes no summary is really needed; a simple list of the summarizing topics is enough. SUMMARIST will print the fuser concepts produced by stage 2 of the process, sorted by decreasing importance.

2. *Phrase concatenation:* SUMMARIST will include a rudimentary generator that composes noun phrase- and clause-sized units into simple sentences. It will extract the noun phrases and clauses from the input text, by following links from the fuser concepts through the words that support them back into the input text.

3. *Full sentence planning and generation:* SUMMARIST will employ the sentence planner being built at ISI (in collaboration with the HealthDoc project from the University of Waterloo) [Hovy and Wanner 96], together with a sentence generator such as Penman [Penman 88, Matthiessen and Bateman 91], FUF [Elhadad 92], or NitroGen [Knight and Hatzivassiloglou 95] to produce well-formed, fluent, summaries, taking as input the fuser concepts and their most closely related concepts as identified by SUMMARIST's topic identification stage.

3 Conclusion

As outlined in Section 1, extract summaries require only the stage of topic identification. By including modules to perform topic interpretation and summary generation, SUMMARIST will also be able to produce abstract summaries. How well it will do so is a matter for future investigation.

An important aspect to be addressed is the combination of the outputs of various modules in each stage. We plan to investigate different

approaches, from a simple combination by votes to methods for automatically training relative strengths of contribution.

Automated summarization is simultaneously an old topic—work on it dates from the 1950's—and a new topic—it is so difficult that interesting headway can be made for many years to come. We are excited about the possibilities offered by the combination of semantic and statistical techniques in what is, quite possibly, the most complex task of all NLP.

References

[Baxendale 58] Baxendale, P.B. 1958. Machine-made index for technical literature—an experiment. IBM Journal (354–361), October.

[Edmundson 69] Edmundson, H.P. 1968. New methods in automatic extraction. In ?, (264–285).

[Elhadad 92] Elhadad, M. 1992. *Using Argumentation to Control Lexical Choice: A Functional Unification-Based Approach*. Ph.D. dissertation, Columbia University.

[Hovy and Wanner 96] Hovy, E.H. and L. Wanner. 1996. Managing Sentence Planning Requirements. In *Proceedings of the Workshop on Planning and Generation* (with ECAI). Budapest, Hungary.

[Jacobs 90] Jacobs, P.S. and L.F. Rau. 1990. SCISOR: Extracting information from on-line news. *Communications of the ACM* 33(11), (88–97).

[Knight and Hatzivassiloglou 95] Knight, K. and V. Hatzivassiloglou. 1995. Two-level many-paths generation. In *Proceedings of the 33rd ACL Conference*, Boston, MA.

[Lin 95] Lin, C.Y. 1995. Topic Identification by Concept Generalization. In *Proceedings of the 33rd ACL Conference*, Boston, MA.

[Lin and Hovy 97a] Lin, C.Y. and E.H. Hovy. 1997a. Identifying Topics by Position. In *Proceedings of the Applied Natural Language Processing Conference*, Washington, DC.

[Lin and Hovy 97b] Lin, C.Y. and E.H. Hovy. 1997b. Automatic Text Categorization: A Concept-Based Approach. In prep.

[Luhn 58] Luhn, H.P. 1959. The automatic creation of literature abstracts. *IBM Journal of Research and Development* (159–165).

[Marcu 97] Marcu, D. 1997. The Rhetorical Parsing of Natural Language Texts. Submitted.

[Matthiessen and Bateman 91] Matthiessen, C.M.I.M. and J.A. Bateman. 1991. *Text Generation and Systemic-Functional Linguistics*. London, England: Pinter

[Mauldin 91] Mauldin, M.L. 1991. *Conceptual Information Retrieval—A Case Study in Adaptive Partial Parsing*. Kluwer Academic Publishers, Boston, MA.

[McKeown and Radev 95] McKeown, K.R. and D.R. Radev. 1995. Generating summaries of multiple news articles. In *Proceedings of the 18th International ACM SIGIR Conference*, (74–82), Seattle, WA.

[Miller et al. 90] Miller, G. R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1990. Five papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University, Princeton, NJ.

[Paice 90] Paice, C.D. 1990. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, 26(1), (171–186).

[Penman 88] *The Penman Primer, User Guide, and Reference Manual*. 1988. Unpublished documentation, USC Information Sciences Institute.

[Salton et al. 94] Salton, G., J. Allen, C. Buckley, and A. Singhal. 1994. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science* 264, (1421–1426), June.

[Schank and Abelson 77] Schank, R.C. and R.P. Abelson. 1977. *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ.

[TREC 95] Harman, D. (ed). 1995. *Proceedings of the TREC Conference*.