Multi-document Summarization via Information Extraction

Michael White and Tanya Korelsky
CoGenTex, Inc., Ithaca, NY
mike,tanya@cogentex.com

Claire Cardie, Vincent Ng, David Pierce, and Kiri Wagstaff
Department of Computer Science, Cornell University, Ithaca, NY
cardie,yung,pierce,wkiri@cs.cornell.edu

1 Introduction

Although recent years has seen increased and successful research efforts in the areas of single-document summarization, multi-document summarization, and information extraction, very few investigations have explored the potential of merging summarization and information extraction techniques. This paper presents and evaluates the initial version of RIPTIDES, a system that combines information extraction (IE), extraction-based summarization, and natural language generation to support user-directed multi-document summarization. We hypothesize that IE-supported summarization will enable the generation of more accurate and targeted summaries in specific domains than is possible with current domain-independent techniques.

In the sections below, we describe the initial implementation and evaluation of the RIPTIDES IE-supported summarization system. We conclude with a brief discussion of related and ongoing work.

2 System Design

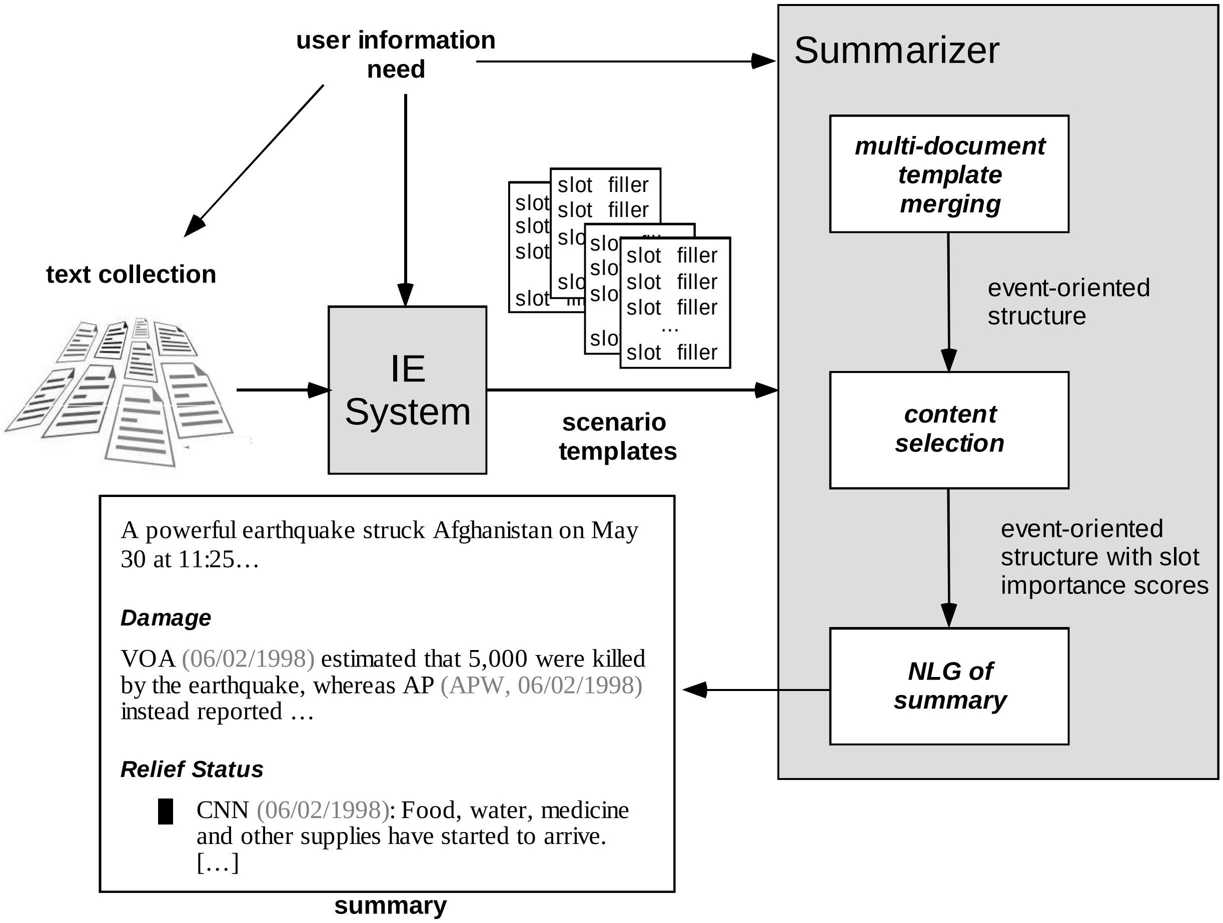
Figure 1 depicts the IE-supported summarization system. The system first requires that the user select (1) a set of documents in which to search for information, and (2) one or more scenario templates (extraction domains) to activate. The user optionally provides filters and preferences on the scenario template slots, specifying what information s/he wants to be reported in the summary. RIPTIDES next applies its Information Extraction subsystem to generate a database of extracted events for the selected domain and then invokes the Summarizer to generate a natural language summary of the extracted information subject to the user's constraints. In the paragraphs below, we describe the IE system and the Summarizer in turn.

IE system. In the RIPTIDES evaluation of Section 3, we assume "perfect" output templates from the IE system. As a result, we focus here on scenario template design considerations that directly affect the quality of the summaries produced rather than the architecture of the IE system,[1] which is of less importance for the purposes of this paper. (We anticipate that the final paper will include both a description of the IE system and an evaluation of the Summarizer using its output.)

The domain selected for the initial system and its evaluation is natural disasters. A top-level natural disasters scenario template contains: document-level information (e.g. *docno*, *date-time*); zero or more *agent* elements denoting each *person*, *group*, and *organization* in the text; and zero or more *disaster* elements. *Agent* elements encode standard information for named entities (e.g. *name*, *position*, *geo-political unit*). For the most part, *disaster* elements also contain standard event-related fields (e.g. *type*,

[1] In brief, the RIPTIDES IE system uses a traditional architecture (Cardie, 1997): a preprocessor finds sentences and tokens; a parser identifies syntactic structure; syntactico-semantic pattern-matching identifies text fragments for extraction; coreference resolution guides template creation. We are investigating the use of weakly supervised learning techniques (e.g. Riloff and Jones, 1999; Thompson et al., 1999) for the automatic construction of each IE system component.

number, *date*, *time*, *location*, *damage* sub-elements). The final product of the RIPTIDES system, however, is not a set of scenario templates, but a user-directed multi-document summary. This difference in goals influenced a number of template design issues. First, disaster elements must distinguish different reports or views of the same event (from multiple sources) — the system creates a separate *disaster* event for each such account — and should include the *reporting agent*, *date*, *time*, and *location* whenever possible. In addition, *damage* elements (i.e. *human* and *physical effects*) are best grouped according to the reporting event. Finally, a slight broadening of the IE task was necessary in that extracted text was not constrained to noun phrases. In particular, adjectival and adverbial phrases that encode *reporter confidence*, and sentences and clauses denoting *relief effort* progress appear beneficial for creating informed summaries. The disaster elements extracted for each text are provided as input to the summarization component.



A powerful earthquake struck Afghanistan on May 30 at 11:25…

Damage

VOA (06/02/1998) estimated that 5,000 were killed by the earthquake, whereas AP (APW, 06/02/1998) instead reported …

Relief Status

- CNN (06/02/1998): Food, water, medicine and other supplies have started to arrive. […]

Figure 1

The Summarizer. A sample summary generated by the initial version of RIPTIDES is shown in Figure 2 below.[2] The Summarizer produces each summary in three main stages. In the first stage, the output

[2] The sample summary was produced for the second evaluation test case (see Section 3) — a summary of all test articles that emphasizes factual information. For space reasons, some of the summary is not shown. In this example, preference is given to the basic facts (the first paragraph), then to the overall damage information (vs. damage reports in specific locations), then to relief information (which has a slight preference over the remaining facts). Note that the relatively poor coherence of the second paragraph stems primarily from the overly simplistic heuristics for comparing damage reports; in the current version of the system, more robust rules are used (see section 4). Also

templates are merged into an event-oriented structure, while keeping track of source information. The merge operation currently relies on simple heuristics to group extracted facts that are comparable; for example, during this phase damage reports are grouped according to whether they pertain to the event as a whole, or instead to damage in the same particular location. Heuristics are also used in this stage to determine the most relevant damage reports, taking into account specificity, recency and news source. With relief slots, word-overlap clustering is used to group slots from different documents into clusters that are likely to report similar content. In the second stage, a base importance score is first assigned to each slot based on a combination of document position, document recency and group/cluster membership. The base importance scores are then adjusted according to user-specified slot preferences and matching criteria. The adjusted scores are used to select the most important slots to include in the summary, subject to the user-specified word limit. In the third and final stage, the summary is generated from the resulting content pool using a combination of top-down, schema-like text building rules and surface-oriented revisions. (At present, however, extracted relief sentences/clauses are simply listed in document order.)

Earthquake strikes northern Afghanistan

A powerful earthquake struck Afghanistan on May 30 at 11:25. The earthquake was centered in a remote part of the country and had a magnitude of 6.9 on the Richter scale.

Damage

VOA (06/02/1998) estimated that 5,000 were killed by the earthquake, whereas AP (APW, 06/02/1998) instead reported anywhere from 2,000 to 5,000 people dead. CNN (06/02/1998) instead reported up to 4, 000 people died, while I (PRI, 06/01/1998) estimated several thousand people may have died. I (PRI, 06/01/1998) estimated that thousands were left homeless. […]

Relief

Status

- CNN (06/02/1998): Food, water, medicine and other supplies have started to arrive. […]

Problems/Obstacles

- VOA (06/03/1998): Bad weather in Afghanistan is hampering efforts to reach victims of last week's devastating earthquake. […]

Further Details

Heavy after shocks shook northern afghanistan. Landslides or mud slides also hit the area. […]

Figure 2

The Summarizer is implemented using the Apache implementation of XSLT (Apache, 2000) and CoGenTex's Exemplars Framework (White and Caldwell, 1998; White, 2001). The Apache XSLT implementation provided a convenient way to rapidly develop a prototype implementation of the first two processing stages using a series of XML transformations. In the first step of the third summary generation stage, the text building component of the Exemplars Framework constructs a "rough draft" of

note that the reference to "I" in this paragraph should have been to the reporter Tony Kahn; our heuristic was to use the initial reference to a source, and "I" happened to be the initial reference in the article.

the summary text.[3] In this rough draft version, XML markup is used to partially encode the rhetorical, referential, semantic and morpho-syntactic structure of the text. In the second generation step, the Exemplars text polishing component makes use of this markup to trigger surface-oriented revision rules that smooth the text into a more polished form. A distinguishing feature of our text polishing approach is the use of a bootstrapping tool to partially automate the acquisition of application-specific revision rules from examples; cf. White (2001) for details.

3 Evaluation and Initial Results

To evaluate the initial version of the IE-supported summarization system, we used Topic 89 from the TDT2 collection — 25 texts on the 1990 Afghanistan earthquake. Each document was annotated manually with the natural disaster scenario templates that comprise the desired output of the IE system. In addition, treebank-style syntactic structure annotations were added automatically using the Charniak (1999) parser. Finally, MUC-style noun phrase coreference annotations were supplied manually. All annotations are in XML.

Next, the Topic 89 texts were split into a development corpus and a test corpus. The development corpus was used to build the summarization system; the evaluation summaries were generated from the test corpus. Summaries generated by the RIPTIDES system were compared to a simple, sentence-extraction multi-document summarizer that relies only on document position, recency, and word overlap clustering. In addition, the RIPTIDES and Baseline system summaries were compared against the summaries of two human authors. All summaries were graded with respect to content, organization, and readability on an A-F scale by four graduate students / professionals, all of whom were unfamiliar with this project.

Each system and author was asked to generate four summaries of different lengths and emphases: (1) a 100-word summary of the May 30 and May 31 articles; (2) a 400-word summary of all test articles, emphasizing specific, factual information; (3) a 200-word summary of all test articles, focusing on the damage caused by the quake, and excluding information about relief efforts, and (4) a 200-word summary of all test articles, focusing on the relief efforts, and highlighting the Red Cross's role in these efforts.

RIPTIDES	Baseline	Person 1	Person 2
C	D/D+	A-	B+

Table 1

	RIPTIDES	Baseline
Overall	1.92 ±0.53	1.16 ±0.48
Content	2.15 ±0.96	1.77 ±1.11
Organization	1.99 ±1.02	0.48 ±0.49
Readability	1.81 ±0.71	1.19 ±0.95

Table 2

The results are shown in Tables 1 and 2. Table 1 provides the overall grade for each system or author averaged across all graders and summaries, where each assigned grade has first been converted to a number (A=4.0, B=3.0, C=2.0, D=1.0, F=0.0) and the average converted back to a letter grade. Table 2

[3] The Exemplars text builder employs a processing model that is similar to XSLT; the primary difference between the two is that the Exemplars text building rules are more object-oriented than XSLT templates, enabling greater rule sophistication and reuse across varying contexts (cf. White and Caldwell, 1998).

shows the mean and standard deviations of the overall, content, organization, and readability scores for the RIPTIDES and the Baseline system averaged across all graders and summaries.

Given the amount of development effort that went into the initial version of the system, we were not surprised that our summarizer fared poorly when compared against manually written summaries, receiving an average grade of C, vs. A- and B+ for the human authors; nevertheless, the initial RIPTIDES system scored almost a full grade ahead of the baseline summarizer, which received a D/D+. The difference in the overall scores was significant, as were the scores for organization and readability (though not content). The most notable improvement was in organization, which was not surprising given that the Baseline system just listed extracted sentences in document order.

The comments of the evaluators helped to identify the most important problems to focus on in ongoing work. These problems include the need for better event description merging, more refined comparison of differences in reported numbers, improved rhetorical structuring of relief information, temporal expression normalization, and sentence reduction. With progress in these areas, we hope to achieve scores within one grade of human performance.

4 Related and Ongoing Work

The RIPTIDES system is most similar to the SUMMONS system of Radev and McKeown (1998), which summarized the results of MUC-4 IE systems in the terrorism domain. In comparison to SUMMONS, the RIPTIDES system appears to be designed to more completely summarize larger input document sets; in particular, we believe our system will scale to handle the hundreds of news articles we have collected about the recent earthquakes in Central America and India, whereas SUMMONS was more of an exploratory prototype that was never run on more than a handful of documents. Another important difference is that SUMMONS sidestepped the problem of comparing reported numbers of varying specificity (e.g. “several thousand” vs. “anywhere from 2000 to 5000” vs. “up to 4000” vs. “5000”), whereas we have recently implemented more robust rules for doing so. In our approach, a range encompassing the current reports across available news sources is constructed, and any lower, less specific or incomparable estimates (e.g. “more than half the region’s residents”) are noted (space permitting).[4]

In its treatment of relief information, the RIPTIDES system is also similar to, though simpler than, the domain-independent multi-document summarizers of Goldstein et al. (2000) and Radev et al. (2000) in the way it clusters sentences across documents to help determine which sentences are central to the collection, as well as to reduce redundancy amongst sentences included in the summary. It is also similar in spirit to MultiGen (Barzilay et al., 2001), though much less ambitious in its approach.

In ongoing work, we are in the process of refining our algorithm for summarizing differences in reported numbers and improving our treatment of relief information. At the conference, we plan on showing output from the current version of the summarizer, using the actual results of the IE system. For the final version of the paper, we plan on repeating our evaluation with the improved system, and will include the updated results in the final version of the paper.

[4] Less specific estimates such as “hundreds” are considered lower than more specific numbers such as “5000” when they are lower by more than a factor of 10.

References

The Apache XML Project. 2001. “Xalan Java.” http://xml.apache.org/.

Barzilay, R., N. Elhadad and K. McKeown. 2001. “Sentence Ordering in Multidocument Summarization.” To appear in the Proceedings of HLT 2001.

Cardie, C. 1997. “Empirical Methods in Information Extraction." *AI Magazine* 18(4): 65-79.

Charniak, E. 1999. “A maximum-entropy-inspired parser.” Brown University Technical Report CS99-12.

Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M. 2000. “Multi-document summarization by sentence extraction.” In *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*, Seattle, WA.

Radev, D. R. and McKeown, K. R. 1998. “Generating natural language summaries from multiple on-line sources.” *Computational Linguistics* 24(3):469-500.

Radev, D. R., Jing, H., and Budzikowska, M. 2000. “Summarization of multiple documents: clustering, sentence extraction, and evaluation.” In *Proceedings of the ANLP/NAACL Workshop on Summarization*, Seattle, WA.

Riloff, E. and Jones, R. 1999. “Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping.” In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, Orlando, FL, pp. 474-479.

Thompson, C. A., Califf, M. A., and Mooney, R. J. 1999. “Active learning for natural language parsing and information extraction.” In *Proceedings of the Sixteenth International Conference on Machine Learning*, Bled, Slovenia, pp. 406-414.

White, M. and Caldwell, T. 1998. “EXEMPLARS: A Practical, Extensible Framework for Dynamic Text Generation.” In *Proceedings of the Ninth International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Canada, pp. 266-275.

White, M. 2001. “Text Polishing: Surface-Oriented Smoothing of Generated Texts via Markup-Based Revision Rules.” In preparation.