An Architecture For Distributed Natural Language Summarization

Dragomir R. Radev

Department of Computer Science, Columbia University
1214 Amsterdam Avenue, New York, NY 10027-7003
radev@cs.columbia.edu

Abstract

We present a system that incorporates agent-based technology and natural language generation to address the problem of natural language summarization of live sources of data. The input to the system includes newswire and on-line databases and ontologies. The output consists of short summaries that convey information selected to fit the user's interests, the most recent news updates, and historical information. The system is under development.

1 Introduction

One of the major problems with the Internet is the abundance of information and the difficulty for the average computer user to read everything existing on a specific topic. There exist now more than 100 sources of live newswire in operation on the Internet. The user has to go through megabytes of news every day to select articles of interest and read the relevant parts of them. Hence, he needs SEARCH AND SELECTION services, as well as for SUMMARIZATION facilities.

There currently exist more than 10 operational SEARCH AND SELECTION services on the Web, e.g., DEC's AltaVista [AltaVista 1996]. However, there is little available in the area of SUMMARIZATION.

The best currently existing Web-based summarization system, *Netsumm* [Preston and Williams 1994], uses a statistical, approach to selecting relevant sentences from an article. It has an impressive user interface, and is practically domain-independent, but suffers from two major problems: it only summarizes a single article at a time, and it only summarizes articles given by the user, which means that the user has to go through hundreds of articles to select the ones he will send to Netsumm.

Other statistical systems [Kupiec *et al.* 1995], [Rau *et al.* 1994] have the same characteristics as Netsumm. Another major unsolved problem involves conveying rapidly changing information to the end user in a sensible format. This information can come from a multitude of different sources which use different internal representations to store it. A summarizing program needs to be able to retrieve all this information in real time, process it and produce meaningful summaries in natural language.

More specifically, the innovations that we suggest address some of these problems.

Asynchronous summarization: Synchronous (demand-based) summarization requires that the user needs to know when a new article relevant to his interests has appeared and *feed it* to the summarizer in order to get a summary back. Such an approach doesn't lead to any economy of time for the user, since he still has to spend time checking whether new articles have been posted and then send them to the summarizer.

It would be more efficient for the user to be notified automatically when a new article has been published [Radev 1994] or to be sent a summary of the article directly. Such asynchronous summaries can be based on the specific interests of the user, contained in his user profile. They can also be tailored to the user's prior knowledge of the subject or event. E.g., the user will receive an initial announcement about an event and only *updates* after that point.

Summarizing multiple articles: All existing statistical summarizers provide summaries of single articles by extracting sentences from them. If such systems were to summarize a series of articles, they would simply process each of them on its own and output the resulting summaries. Such summaries will likely contain a significant amount of repeated information, as do the source articles themselves.

Our summarizer works on a *set* of articles. It can trace the development of an event over time or contradictions in articles from different sources on the same topic.

Summarizing multiple sources: When different sources present exactly the same information, the user clearly needs only have access to one of them. Practically, this assumption doesn't hold, as different sources provide updates from a different perspective and at dif-

ferent times. An intelligent summarizer's task, therefore, would be to attain as much information from the multiple sources as possible, combine it, and present it in a concise form to the user. For example, if two sources of information report a different number of casualties in a particular incident, the summarizer will report the contradiction.

Symbolic summarization: An inherent problem to sentence-extraction based summarizers is the lack of fluency in the output. The extracted sentences fit together only in the case they are adjacent in the source document. It is also clear that these sentences weren't meant to serve as summaries. A system that provides a deeper understanding of the message (or set of messages) will have all necessary information to get a fluent surface summary.

Interoperability: Since a large-scale summarization system should monitor multiple sources of news and other data, it has to use a knowledge transmission language in order to coordinate the multiple autonomous sources.

In the following section, we will describe our early summarization prototype, SUMMONS [McKeown and Radev 1995]. In the next sections, we will describe our architecture for real-time summarization, as well as our approach to the issues set forth in the current section.

2 SUMMONS

Our choice of domain was dictated by the existence of two Message Understanding Conferences (MUC) organized by DARPA [Sundheim 1992] in the domain of terrorism in Latin America. The participants were asked to fill templates (as shown in Figure 1) with information extracted from news articles. We parsed the templates (Figure 2), adding information about the primary and secondary sources of news[1].

SUMMONS (SUMMarizing Online NewS articles) is based on an architecture used in PLANDoc [McKeown *et al.* 1994], developed jointly by Bellcore and Columbia University. It consists of a content planner which decides what information is to be included in the summary, and a surface generator, based on the FUF/SURGE tools developed by Michael Elhadad [Elhadad 1993]. We have used SUMMONS on templates from two MUC conferences (covering events in 1988 and 1989) and on manually generated templates from recent events (e.g., the 1993 World Trade Center bombing).

SUMMONS (Figure 3) uses summarization operators to express various ways in which the templates that are to be generated are related to each other. We have implemented operators for *Superset*, *Addition*, *Contradiction*, *Refinement*, *Change of Perspective*, etc. The following paragraph was generated by the *Change of Perspective* operator on a set of two messages.

> The afternoon of February 26, 1993, Reuters reported that a suspected bomb killed at least five people in the World Trade Center. Later the same day, Reuters announced that exactly five people were killed in the blast.

MESSAGE: ID	TST3-MUC4-0010
INCIDENT: DATE	01 NOV 89
INCIDENT: LOCATION	EL SALVADOR
INCIDENT: TYPE	ATTACK
INCIDENT: STAGE OF EXECUTION	ACCOMPLISHED
INCIDENT: INSTRUMENT TYPE	-
PERP: INCIDENT CATEGORY	TERRORIST ACT
PERP: INDIVIDUAL ID	"TERRORIST"
PERP: ORGANIZATION ID	"THE FMLN"
PERP: ORG. CONFIDENCE	REPORTED: "THE FMLN"
HUM TGT: TYPE	CIVILIAN: "1 CIVILIAN"
HUM TGT: NUMBER	1: "1 CIVILIAN"
HUM TGT: EFFECT OF INCIDENT	DEATH: "1 CIVILIAN"

Figure 1: Excerpts from a MUC-4 Template.

```
(message
  (system (id "TST3-MUC4-0010"))
  (source (secondary "NCCOSC"))
  (incident (date "01 NOV 89")
          (location "El Salvador")
          (type attack)
          (stage accomplished))
  (perpetrator (category terr-act)
              (org-id "THE FMLN")
              (org-conf rep-fact))
  (victim (description civilian)
         (number 1)
)
```

Figure 2: Parsed MUC-4 Template.

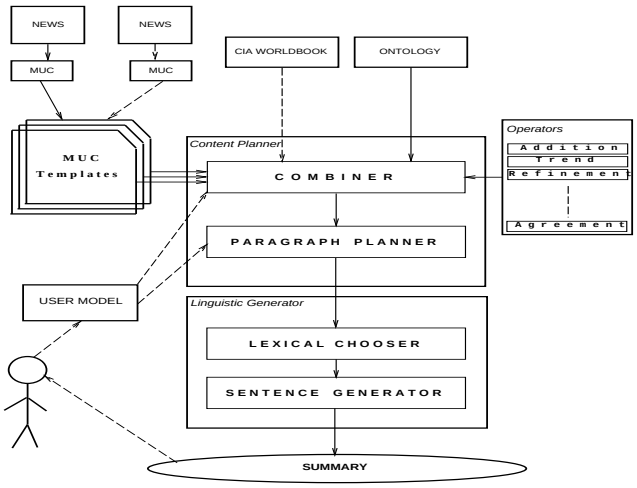

Figure 3: SUMMONS System Architecture.

3 Summarization architecture

The interoperability problem is addressed using a proposed standard for exchange of information and knowledge - KQML [Finin *et al.* 1994]. KQML aims at

[1] The primary source, e.g., an eyewitness, and the secondary source, e.g., a news agency, are very important for producing accurate summaries

the standardization of both a protocol and a message format for communication among independent processes over a wide-area network. KQML is used to create facilitators which provide the interface between heterogeneous applications which run on various machines and which are written in various programming languages. Such facilitators communicate through KQML performatives and exchange messages written in some content language. In our case, this is a simple template language, developed locally.

Our architecture draws from work on Software Agents [Genesereth and Ketchpel 1994]. Our goal was to expand the model to incorporate natural language interfaces. We have used agents of various types in a modular way:

```
(country (name "El Salvador")
  (capital "San Salvador")
  (map (url
  "http://www.odci.gov/cia/publications/95fact/es.gif"))
  (divisions (name "department")
    (list ("Ahuachapan"
           ...
          "Usulutan")))
  (executive (president
              (name "Armando CALDERON SOL")
              (elected "010694")))
)
```

Figure 4: Parsed World Book entry.

Summarizer(s): agents that are concerned with summarizing the data that they have collected over the network from different sources and producing natural-language reports for the end-user. The summarizer is connected with the user model and the user interface.

Database servers: expert agents that have access to knowledge bases which are updated periodically and which contain information that is less likely to change over the course of a summarization session (e.g. heads of state, geographical and common-sense knowledge). In our case, such information comes from two sources: the CIA World Book [CIA 1995] and the ontologies supplied with the MUC conferences. An example from the World Book related to El Salvador is shown in Figure 4. The World Book facilitator parses the entries for each country into a Lisp-like format and provides access to them to the planner. Another instance of a database server is the facilitator connected to the node labeled *Ontology* in Figure 3. This represents the database containing the ontologies (including geographical locations, weapons, and incident types, available from the MUC conference).

Data collectors: agents that are connected to the real world through filters or use human experts who can feed real-time raw data such as sports scores, news updates, changes in stock prices, etc. They are connected to the rest of the modules through the intermediary of facilitators that convert from the template format to KQML and vice-versa. In our system, the role of data collectors is performed by the MUC systems and the facilitators connected to the World Book.

Planner: it maintains contacts with the facilitators in order to keep the knowledge base of the summarizer up to date. It uses KQML subscription messages to learn in an asynchronous way about changes in the knowledge bases of other facilitators.

The following example shows how the planner uses a KQML subscription message to subscribe to new messages related to El Salvador.

```
(subscribe
  :content
    (EQ
      (message
        (incident
          (location "El Salvador"))))
  :ontology geog-onto
  :language KQML
  :reply-with "loc-salvador-1"
  :sender "planner"
  :receiver "muc1"
)
```

Whenever a new message becomes available (E.g., Figure 2), the MUC facilitator will reply with an appropriate message.

```
(reply
  :content
    (message
      (system
        (id "TST3-MUC4-0010")
        ...
    ))
  :ontology geog-onto
  :language KQML
  :in-reply-to "loc-salvador-1"
  :sender "muc1"
  :receiver "planner"
)
```

Other KQML performatives, such as *ask-all*, *ask-one*, *register*, *tell*, or *sorry* have also been implemented.

User Model: it keeps information about the user's interests (e.g. keywords, regions in the world), preferences (how frequently he wants to get updates), and interaction history (what information has already been shown to him). Let's consider the case in which the user has already been notified about a terrorist act:

> A bombing took place on August 23rd, 1988 in the district of Talcahuano, Chile.

The next time the system needs to refer to the same event, it can omit some information that it has already shown to the user (e.g., the fact that Talcahuano is in Chile), and can instead focus on information that has not been included previously.

> *The Talcahuano bombing* didn't result in any injuries. However, the *Chapel of the Church of Jesus* was damaged.

4 Current Work and Directions for Future Research

Currently, our system can handle simple summaries consisting of 1-3 sentence paragraph which are limited to the MUC domain and to a few additional events for which we have manually created MUC-like templates. Several components related to interoperability are also fully implemented (e.g., the subscription package in KQML and the query-response interface to the MUC and World Book facilitators). We haven't yet connected the system to a working MUC component[2]. The user model hasn't been implemented yet.

A problem that we haven't addressed is related to the clustering of articles according to their relevance to a specific event. Another issue is domain-independence.

Since the understanding and generation modules share only language-independent templates, we would try to implement a limited form of machine translation by summarizing in one language news written in another language.

5 Conclusions

We have described an agent-based system which allows for summarization of multiple articles from multiple sources in an asynchronous fashion while taking into account user preferences. We have also shown how such an architecture can be modular and extensible and how its different components interact.

Acknowledgments:

I would like to thank my adviser, Prof. Kathleen McKeown, and also James Shaw and Karen Kukich for the interaction on PLANDoc, and Evelyne Tzoukermann for help with reviewing a version of this paper.

References

Altavista. WWW site, URL: http:// altavista.digital.com, 1996.

CIA. The CIA World Factbook. URL: http:// www.odci.gov/cia/publications/95fact, 1995.

Michael Elhadad. *Using argumentation to control lexical choice: a unification-based implementation.* PhD thesis, Computer Science Department, Columbia University, 1993.

Tim Finin, Rich Fritzson, Don McKay, and Robin McEntire. KQML - A Language and Protocol for Knowledge and Information Exchange. Technical Report CS-94-02, Computer Science Department, University of Maryland and Valley Forge Engineering Center, Unisys Corporation, 1994.

Michael Genesereth and Steven Ketchpel. Software Agents. *Communications of the ACM*, 37(7):48–53, July 1994.

Julian M. Kupiec, Jan Pedersen, and Francine Chen. A Trainable Document Summarizer. In *Proceedings, 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, Seattle, Washington, July 1995.

Kathleen R. McKeown and Dragomir R. Radev. Generating Summaries of Multiple News Articles. In *Proceedings, 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–82, Seattle, Washington, July 1995.

Kathleen R. McKeown, Karen Kukich, and James Shaw. Practical Issues in Automatic Documentation Generation. In *Proceedings of the ACL Applied Natural Language Conference*, Stuttgart, Germany, October 1994.

Keith Preston and Sandra Williams. Managing the Information Overload. *Physics in Business*, June 1994.

Dragomir R. Radev. Rendezvous: A WWW Synchronization System. Poster Session, Second International WWW Conference, Chicago, Illinois, October 1994.

L.F. Rau, R. Brandow, and K. Mitze. Domain-Independent Summarization of News. In *Summarizing Text for Intelligent Communication*, pages 71–75, Dagstuhl, Germany, 1994.

Beth M. Sundheim. Overview of the fourth message understanding evaluation and conference. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 3–21, McLean, Virginia, June 1992.

[2] We are in the process of acquiring working MUC systems from NYU and BBN.