Automatic Text Summarization Based on the Global Document Annotation

Katashi Nagao
Sony Computer Science Laboratory Inc.
3–14–13 Higashi-gotanda, Shinagawa–ku,
Tokyo 141-0022, Japan
nagao@csl.sony.co.jp

Kôiti Hasida
Electrotechnical Laboratory
1–1–4 Umezono, Tukuba,
Ibaraki 305-8568, Japan
hasida@etl.go.jp

Abstract

The GDA (Global Document Annotation) project proposes a tag set which allows machines to automatically infer the underlying semantic/pragmatic structure of documents. Its objectives are to promote development and spread of NLP/AI applications to render GDA-tagged documents versatile and intelligent contents, which should motivate WWW (World Wide Web) users to tag their documents as part of content authoring. This paper discusses automatic text summarization based on GDA. Its main features are a domain/style-free algorithm and personalization on summarization which reflects readers' interests and preferences. Our solution naturally outperforms the traditional summarization methods, which just pick out sentences highly scored on the basis of superficial clues such as word count, etc. In order to calculate the importance score of a text element, the algorithm uses spreading activation on an intra-document network which connects text elements via thematic, rhetorical, and coreferential relations. The proposed method is flexible enough to dynamically generate summaries of various sizes. A summary browser supporting personalization is reported as well.

1 Introduction

The WWW has opened up an era in which an unrestricted number of people publish their messages electronically through their online documents. However, it is still very hard to automatically process contents of those documents. The reasons include the following:

1. HTML (HyperText Markup Language) tags mainly specify the physical layout of documents. They address very few content-related annotations.
2. Hypertext links cannot very much help readers recognize the content of a document.
3. The WWW authors tend to be less careful about wording and readability than in traditional printed media. Currently there is no systematic means for quality control in the WWW.

Although HTML is a flexible tool that allows you to freely write and read messages on the WWW, it is neither very convenient to readers nor suitable for automatic processing of contents.

We have been developing an integrated platform for document authoring, publishing, and reuse by combining natural language and WWW technologies. As the first step of our project, we defined a new tag set and developed tools for editing tagged texts and browsing these texts. The browser has the functionality of summarization and content-based retrieval of tagged documents.

This paper focuses on summarization based on this system. The main features of our summarization method are a domain/style-free algorithm and personalization to reflect readers' interests and preferences. This method naturally outperforms the traditional summarization methods, which just pick out sentences highly scored on the basis of superficial clues such as word count, and so on.

In the rest of this paper, we briefly describe our project called GDA (Global Document Annotation), then discuss our summarization method and personalization on summarization using an implemented prototype.

2 Global Document Annotation

GDA (Global Document Annotation) is a challenging project to make WWW texts machine-understandable on the basis of a new tag set, and to develop content-based presentation, retrieval, question-answering, summarization, and translation systems with much higher quality than before. GDA thus proposes an integrated global platform for electronic content authoring, presentation, and reuse.

The GDA tag set is based on XML (Extensible Markup Language), and designed as compatible as possible with HTML, TEI, EAGLES, and so forth. It specifies modifier-modifiee relations,

anaphor-referent relations, word senses, etc. An example of a GDA-tagged sentence is as follows:

```
<su><np sem=time0>time</np>
<vp><v sem=fly1>flies</v>
<adp><ad sem=like0>like</ad> <np>an
<n sem=arrow0>arrow</n></np>
</adp></vp>.</su>
```

`<su>` means sentential unit.
`<n>`, `<np>`, `<v>`, `<vp>`, `<ad>` and `<adp>` mean noun, noun phrase, verb, verb phrase, adnoun or adverb (including preposition and postposition), and adnominal or adverbial phrase, respectively[1].

The GDA initiative aims at having many WWW authors annotate their on-line documents with this common tag set so that machines can automatically recognize the underlying semantic and pragmatic structures of those documents much more easily than by analyzing traditional HTML files. A huge amount of annotated data is expected to emerge, which should serve not just as tagged linguistic corpora but also as a worldwide, self-extending knowledge base, mainly consisting of examples showing how our knowledge is manifested.

GDA has three main steps:

1. Propose an XML tag set which allows machines to automatically infer the underlying structure of documents.
2. Promote development and spread of NLP/AI applications to turn tagged texts to versatile and intelligent contents.
3. Motivate thereby the authors of WWW files to annotate their documents using those tags.

The tags proposed in Step 1 will also encode coreferences, rhetorical structure, the social relationship between the author and the audience, etc., in order to render the document machine-understandable.

Step 2 concerns AI applications such as machine translation, information retrieval, information filtering, data mining, consultation, expert systems, and so on. If annotation with such tags as mentioned above may be assumed, it is certainly possible to drastically improve the accuracy of such applications. New types of applications for communication aids may be invented as well.

Step 3 encourages WWW authors to present themselves to the widest and best possible audience by organized tagging. WWW authors will be motivated to annotate their Web pages, because documents annotated according to a common standard can be translated, retrieved, etc., with higher accuracy, and thus have a greater chance to reach more targeted readers. Thus, tagging will make documents stand out much more effectively than decorating them with pictures and sounds.

[1] A more detailed description of the GDA tag set can be found at `http://www.etl.go.jp/etl/nl/GDA/tagset.html`.

2.1 Thematic/Rhetorical Relations

The `rel` attribute encodes a relationship in which the current element stands with respect to the element that it semantically depends on. Its value is called a relational term. A relational term denotes a binary relation, which may be a thematic role such as agent, patient, recipient, etc., or a rhetorical relation such as cause, concession, etc. Thus we conflate thematic roles and rhetorical relations here, because the distinction between them is often vague. For instance, *concession* may be both intrasentential and intersentential relation.

Here is an example of a `rel` attribute:

```
<su syn=f><name rel=agt>Tom</name>
<vp>came</vp>.</su>
```

`syn=f` means that the first element `<name rel=agt>Tom</name>` depends on the second element `<vp>came</vp>`. `rel=agt` means that Tom has the agent role with respect to the event denoted by *came*.

`rel` is an open-class attribute, potentially encompassing all the binary relations lexicalized in natural languages. An exhaustive listing of thematic roles and rhetorical relations appears impossible, as widely recognized. We are not yet sure about how many thematic roles and rhetorical relations are sufficient for engineering applications. However, the appropriate granularity of classification will be determined by the current level of technology.

2.2 Anaphora and Coreference

Each element may have an identifier as the value of the `id` attribute. Anaphoric expression should have the `ana` attribute with its antecedent's `id` value. An example follows:

```
<name id=1>John</name> beats
<adp ana=1>his</adp> dog.
```

A non-anaphoric coreference is marked by the `crf` attribute, whose usage is the same as the `ana` attribute.

When the coreference is at the level of type (kind, sort, etc.) which the referents of the antecedent and the anaphor are tokens of, we use the `ctp` attribute as below:

```
You bought <np id=11>a car</np>.
I bought <np ctp=11>one</np>, too.
```

A zero anaphora is encoded by using the appropriate relational term as an attribute name with the referent's `id` value. Zero anaphors of compulsory elements, which describe the internal structure of the events represented by the verbs of adjectives are required to be resolved. Zero anaphors of optional elements such as with reason and means roles may not. Here is an example of a zero anaphora concerning an optional thematic role `ben` (for *beneficiary*):

```
Tom visited <name id=111>Mary</name>.
He <v ben=111>brought</v> a present.
```

3 Text Summarization

As an example of a basic application of GDA, we have developed an automatic text summarization system. Summarization generally requires deep semantic processing and a lot of background knowledge. However, most previous works use several superficial clues and heuristics on specific styles or configurations of documents to summarize.

For example, clues for determining the importance of a sentence include (1) sentence length, (2) keyword count, (3) tense, (4) sentence type (such as fact, conjecture and assertion), (5) rhetorical relation (such as reason and example), and (6) position of sentence in the whole text. Most of these are extracted by a shallow processing of the text. Such a computation is rather robust.

Present summarization systems (Watanabe, 1996; Hovy and Lin, 1997) use such clues to calculate an importance score for each sentence, choose sentences according to the score, and simply put the selected sentences together in order of their occurrences in the original document. In a sense, these systems are successful enough to be practical, and are based on reliable technologies. However, the quality of summarization cannot be improved beyond this basic level without any deep content-based processing.

We propose a new summarization method based on GDA. This method employs a spreading activation technique (Hasida et al., 1987) to calculate the importance values of elements in the text. Since the method does not employ any heuristics dependent on the domain and style of documents, it is applicable to any GDA-tagged documents. The method also can trim sentences in the summary because importance scores are assigned to elements smaller than sentences.

A GDA-tagged document naturally defines an intra-document network in which nodes correspond to elements and links represent the semantic relations mentioned in the previous section. This network consists of sentence trees (syntactic head-daughter hierarchies of subsentential elements such as words or phrases), coreference/anaphora links, document/subdivision/paragraph nodes, and rhetorical relation links.

Figure 1 shows a graphical representation of the intra-document network.

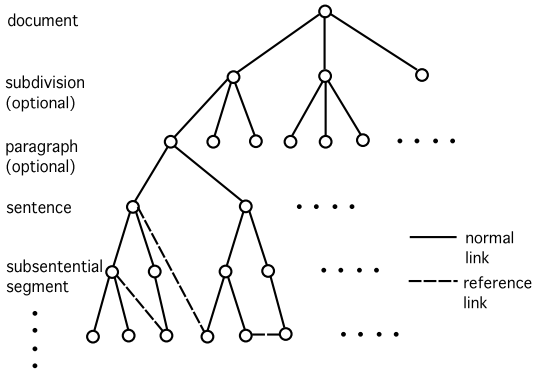

Figure 1: Intra-Document Network

The summarization algorithm is the following:

1. Spreading activation is performed in such a way that two elements have the same activation value if they are coreferent or one of them is the syntactic head of the other.
2. The unmarked element with the highest activation value is marked for inclusion in the summary.
3. When an element is marked, other elements listed below are recursively marked as well, until no more element may be marked.
   - its head
   - its antecedent
   - its compulsory or *a priori* important daughters, the values of whose relational attributes are `agt`, `pat`, `obj`, `pos`, `cnt`, `cau`, `cnd`, `sbm`, and so forth.
   - the antecedent of a zero anaphor in it with some of the above values for the relational attribute
4. All marked elements in the intra-document network are generated preserving the order of their positions in the original document.
5. If a size of the summary reaches the user-specified value, then terminate; otherwise go back to Step 2.

The following article of the Wall Street Journal was used for testing this algorithm.

> During its centennial year, The Wall Street Journal will report events of the past century that stand as milestones of American business history. THREE COMPUTERS THAT CHANGED the face of personal computing were launched in 1977. That year the Apple II, Commodore Pet and Tandy TRS came to market. The computers were crude by today's standards. Apple II owners, for example, had to use their television sets as screens and stored data on audiocassettes. But Apple II was a major advance from Apple I, which was built in a garage by Stephen Wozniak and Steven Jobs for hobbyists such as the Homebrew Computer Club. In addition, the Apple II was an affordable $1,298. Crude as they were, these early PCs triggered explosive product development in desktop models for the home and office. Big mainframe computers for business had been around for years. But the new 1977 PCs – unlike earlier built-from-kit types such as the Altair, Sol and IMSAI – had keyboards and could store about two pages of data in their memories. Current PCs are more than 50 times faster and have memory capacity 500 times greater than their 1977 counterparts. There were many pioneer PC contributors. William Gates and Paul Allen in 1975 developed an early language-housekeeper system for PCs, and Gates became an industry billionaire six years after IBM adapted one of these versions in 1981. Alan F. Shugart, currently chairman of Seagate Technology, led the team that developed the disk drives for PCs. Dennis Hayes and Dale Heatherington, two Atlanta engineers, were co-developers of the internal modems that allow PCs to share data via the telephone. IBM, the world leader in computers, didn't offer its first PC until August 1981 as many other companies entered the market. Today, PC shipments annually total some $38.3 billion worldwide.

Here is a short, computer-generated summary of this sample article:

> THREE COMPUTERS THAT CHANGED the face of personal computing were launched. Crude as they were, these early PCs triggered explosive product development. Current PCs are more than 50 times faster and have memory capacity 500 times greater than their counterparts.

The proposed method is flexible enough to dynamically generate summaries of various sizes. If a longer summary is needed, the user can change the window size of the summary browser, as described in Section 3.1. Then, the summary changes its size to fit into the new window. An example of a longer summary follows:

> THREE COMPUTERS THAT CHANGED the face of personal computing were launched. The Apple II, Commodore Pet and Tandy TRS came to market. The computers were crude. Apple II owners had to use their television sets and stored data on audiocassettes. The Apple II was an affordable $1,298. Crude as they were, these early PCs triggered explosive product development. The new PCs had keyboards and could store about two pages of data in their memories. Current PCs are more than 50 times faster and have memory capacity 500 times greater than their counterparts. There were many pioneer PC contributors. William Gates and Paul Allen developed an early language-housekeeper system, and Gates became an industry billionaire after IBM adapted one of these versions. IBM didn't offer its first PC.

An observation obtained from this experiment is that tags for coreferences and thematic and rhetorical relations are almost enough to make a summary. In particular, coreferences and rhetorical relations help summarization very much.

GDA tags allow us to apply more sophisticated natural language processing technologies to come up with better summaries. It is straightforward to incorporate sentence generation technologies to paraphrase parts of the document, rather than just selecting or pruning them. Annotations on anaphora can be exploited to produce context-dependent paraphrases. Also the summary could be itemized to fit in a slide presentation.

3.1 Summary Browser

We developed a summary browser using a Java-capable WWW browser. Figure 2 shows an example screen of the summary browser.

It has the following functionalities:

1. A screen is divided into three parts (frames). One frame provides a user input form through

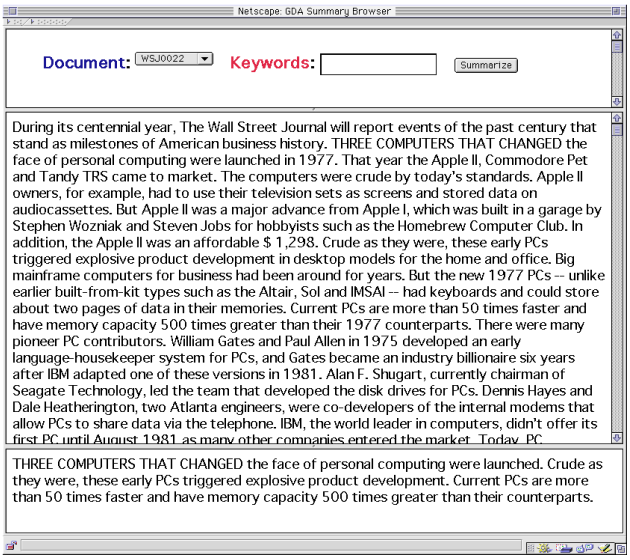

Figure 2: Summary Browser

which you can select documents and type keywords. The other frames are for displaying the original document and its summary.

2. The frame for the summary text is resizable by sliding the boundary with the original document frame. The size of the summary frame influences the size of the summary itself. Thus you can see the summary in a preferred size and change the size in an easy and intuitive way, as shown in Figure 3.

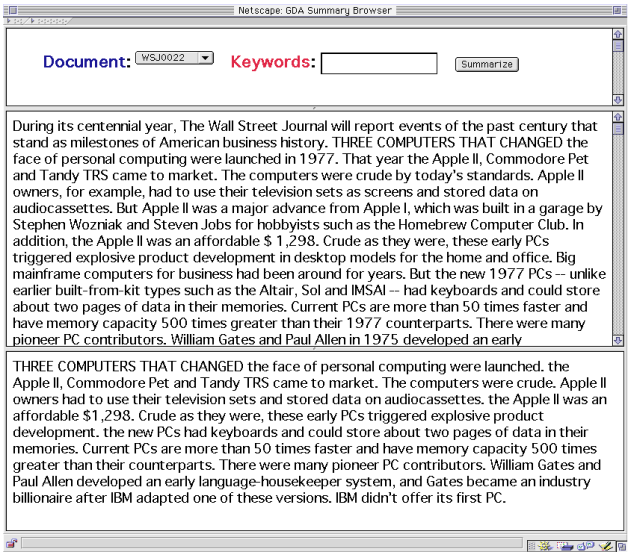

Figure 3: Resized Summary Frame

3. The frame for the original document is mouse sensitive. You can select any element of text in this frame. This function is used for the customization of the summary, as described later. Figure 4 shows that the user selected word 'IBM' on the original document frame. Then, the summary is updated by reperforming spreading activation.

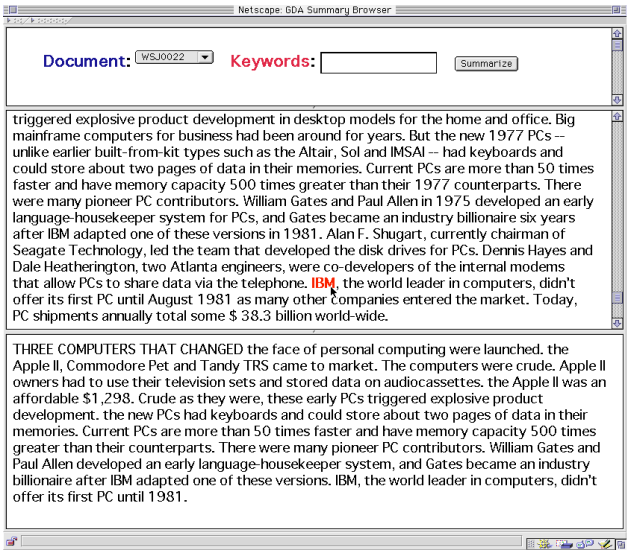

Figure 4: Selection of Word on the Original Document Frame

4. HTML tags are also handled by the browser. So, images are viewed and hyperlinks are managed both in the summary. If a hyperlink is clicked in the original document frame, the linked document appears on the same frame. The hyperlinks are kept in the summary.

4 Personalization

A good summary might depend on the background knowledge of its creator. It also should change according to the interests or preferences of its reader. Let us refer to the adaptation of the summarization process to a particular user as *personalization*. GDA-based summarization can be easily personalized because our method is flexible enough to bias a summary toward the user's concerns. You can select any elements in the original document during summarization, to interactively provide information concerning your personal interests.

We have been developing the following techniques for personalized summarization:

- Keyword-based customization
  The user can input any words of interest. The system relates those words with those in the document using cooccurrence statistics acquired from a corpus and a dictionary such as

WordNet (Miller, 1995). The related words in the document are assigned numeric values that reflect closeness to the input words. These values are used in spreading activation for calculating importance scores.

- Interactive customization by selecting any elements from a document

  The user can mark any words, phrases, and sentences to be included in the summary. The summary browser allows the user to select those elements by pointing devices such as mouse and stylus pen. The user can easily select elements by clicking on them. The click count corresponds to the level of elements. That is, the first click means the word, the second the next larger element containing it, and so on. The selected elements will have higher activation values in spreading activation.

- Learning user interests by observation of WWW browsing

  The summarization system can customize the summary according to the user without any explicit user inputs. We implemented a learning mechanism for user personalization. The mechanism uses a weighted feature vector. The feature corresponds to the category or topic of documents. The category is defined according to a WWW directory such as Yahoo. The topic is detected using the summarization technique.

  Learning is roughly divided into data acquisition and model modification. The user's behavioral data is acquired by detecting her information access on the WWW. This data includes the time and duration of that information access and features related to that information. The first step of model modification is to estimate the degree of relevance between the input feature vector assigned to the information accessed by the user and the model of the user's interests acquired from previous data. The second step is to adjust the weights of features in the user model.

  The model modification algorithm is very simple, because we calculate the average value of all feature vectors. The reason is as follows: Let x be a model of interests, and $\{e_1, e_2, ..., e_n\}$ be a set of feature vectors. A relevance value between a feature vector e_1 and model x is given by the inner product of the two vectors $e_i * x$. Then, in order to maximize the sum of relevance values $S(x) = \Sigma_i(e_i * x) = nE * x$, x should be αE, where E is the avarage of all feature vectors and α is a positive constant for normalization.

5 Concluding Remarks

We have discussed the GDA project, which aims at supporting versatile and intelligent contents. Our focus in the present paper is one of its applications to automatic text summarization. We are evaluating our summarization method using online Japanese articles with GDA tags. We are also extending text summarization to that of hypertext. For example, a summary of a hypertext document will include recursively embedding linked documents in summary, which should be useful for encyclopedic entries, too.

Future work includes construction of a large-scale GDA corpus and system evaluation by open experimentation. GDA tools including a tagging editor and a browser will soon be publicly available on the WWW. Our main current concern is interactive and intelligent presentation, as an extension of text summarization. This may turn out to be a killer application of GDA, because it does not just presuppose rather small amount of tagged document but also makes the effect of tagging immediately visible to the author. We hope that our project revolutionize global and intercultural communications.

Acknowledgments

The authors would like to thank Masao Utiyama and Christoph Neumann for discussion about GDA. Special thanks go to the students of Ishizaki Laboratory at Keio University, especially Jun Okamoto, Hiroaki Teraoka, and Natsuko Hitachi, for their help in developing GDA-tagged documents and tools for them.

References

Kôiti Hasida, Syun Ishizaki, and Hitoshi Isahara. 1987. A connectionist approach to the generation of abstracts. In Gerard Kempen, editor, *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, pages 149–156. Martinus Nijhoff.

Eduard Hovy and Chin Yew Lin. 1997. Automated text summarization in SUMMARIST. In *Proceedings of ACL Workshop on Intelligent Scalable Text Summarization.*

George Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Hideo Watanabe. 1996. A method for abstracting newspaper articles by using surface clues. In *Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING-96)*, pages 974–979.