Automatic Slide Presentation from Semantically Annotated Documents

UTIYAMA Masao
Shinshu University
mutiyama@sp.shinshu-u.ac.jp

HASIDA Kôiti
Electrotechnical Laboratory
hasida@etl.go.jp

Abstract

This paper discusses how to automatically generate slide shows. The reported presentation system inputs documents annotated with the GDA tagset, an XML tagset which allows machines to automatically infer the semantic structure underlying the raw documents. The system picks up important topics in the input document on the basis of the semantic dependencies and coreferences identified from the tags. This topic selection depends also on interactions with the audience, leading to dynamic adaptation of the presentation. A slide is composed for each topic by extracting relevant sentences and paraphrasing them to an itemized summary. Some heuristics are employed here for paraphrasing and layout. Since the GDA tagset is independent of the domain and style of documents and applicable to diverse natural languages, the reported system is also domain/style independent and easy to adapt to different languages.

1 Introduction

A presentation of information content must be adapted to the context. A problem arises here because of diverse types of contexts mainly due to the audience's idiosyncratic needs, backgrounds, and so forth. Adaptation by learning [Perkovitz and Etzioni, 1997; 1998] cannot provide a full solution here, because individual information seekers' profiles and contexts are unpredictable from past experiences. It is essentially necessary to dynamically customize a presentation through interactions with the audience, as human presenters normally do.

In the present paper we discuss how to automatically generate slide shows from semantically annotated documents, in such a way that the presentation can be dynamically adapted to the audience. The reported presentation system detects important topics in the input document and composes a slide for each topic by extracting and paraphrasing relevant sentences. This whole process takes into consideration not only the semantic structure of the given document but also interactions with the audience. So the slide show can be dynamically customized by reflecting requests and queries from the audience during the presentation.

Each slide is typically an itemized summary of a topic in the original document. Generating such slides and coordinating them to meet the audience's needs involves a lot more drastic reformation of the original document than mere extraction of sentences in traditional summarization, so that accurate semantic structure of the document is necessary. We hence assume that the input documents come with GDA (Global Document Annotation) tags [Hasida, 1997; Nagao and Hasida, 1998] embedded. The GDA tagset is an XML (eXtensible Markup Language) tagset which allows machines to automatically infer the semantic structures (including pragmatic structures) underlying the raw documents.

Under the current state of the art, GDA-tagging can be only semiautomatic and calls for manual correction. The cost involved here pays, because an annotated document is a generic form of information content from which to compose diverse types of presentations, potentially involving summarization, narration, visualization, translation, information retrieval, information extraction, and so forth. The slide presentation system reported below addresses a core technology in this broad setting. In the rest of the paper, we first outline the GDA tagset, and discuss how to extract topics from the input document and to generate slides for them by exploiting the tags.

2 The GDA Tagset

GDA is a project to make WWW texts machine-understandable on the basis of a linguistic tag set, and to develop applications such as content-based presentation, retrieval, question-answering, summarization, and translation with much higher quality than before. GDA thus proposes an integrated global platform for electronic content authoring, presentation, and reuse. The GDA tagset[1] is based on XML, and designed as compatible as possible with HTML, and TEI[2], etc., incorporat-

[1] `http://www.etl.go.jp/etl/nl/GDA/tagset.html`
[2] http://www.uic.edu:80/orgs/tei/

ing insights from EAGLES[3], Penn TreeBank [Marcus *et al.*, 1993], and so forth.

Described below is a minimal outline of the GDA tagset necessary for the rest of the discussion. Parse-tree bracketing, semantic relation, and coreference are essential for slide presentation, as with many other applications such as translation. Further details, concerning coordination, scoping, illocutionary act, and so on, are omitted.

2.1 Parse-Tree Bracketing

As the primary purpose of GDA tagging is to encode semantic structure, syntactic annotation is exploited only as far as it contributes to semantic encoding. Also, syntactic tags are designed to simplify syntactic annotation by minimizing the number of tags and accordingly the depth of embedding among them.

An example of a GDA-tagged sentence is shown in Figure 1. <su> means sentential unit. <np>, <v>, and

```
<su>
  <np rel="sbj">time</np>
  <v>flies</v>
  <adp>
    like
    <np>an arrow</np>
  </adp>
  .
</su>
```

Figure 1: A GDA-tagged sentence.

<adp> stand for noun phrase, verb, and adnominal or adverbial phrase.

<su> and the tags whose name end with 'p' (such as <adp> and <vp>) are called *phrasal tags*. In a sentence, an element (a text span enclosed in a begin tag and the corresponding end tag) is usually a syntactic constituent. The elements enclosed in phrasal tags are *phrasal elements*, which cannot be the head of larger elements. So in Figure 1 'flies' is specified to be the head of the <su> element and 'like' the head of the <adp> element.

2.2 Semantic Relation

The `rel` attribute encodes a relationship in which the current element stands with respect to the element that it syntactically depends on. Its value represents a binary relation, which may be a grammatical function such as SUBJECT, a thematic role such as AGENT, PATIENT, RECIPIENT, or a rhetorical relation such as CAUSE, CONCESSION, and ELABORATION. Grammatical functions are used to encode semantic relation assuming that a dictionary is available by which to associate grammatical functions with thematic roles for lexical items such as verbs. Thematic roles and rhetorical relations are also conflated, because the distinction between them is often vague. For instance, CONCESSION may be both intrasentential and intersentential relation.

[3]http://www.ilc.pi.cnr.it/EAGLES/home.html

2.3 Coreference

As discussed later, coreferences play a major role in slide presentation. `id`, `eq`, `ctp`, `sub` and `sup` attributes are mainly used to encode coreferences. Each element may have an identifier as the value for the `id` attribute. Coreferent expression should have the `eq` attribute with its antecedent's `id` value. An example follows:

```
<np id="j0">John</np> beats
<adp eq="j0">his</adp> dog.
```

When the shared semantic content is not the referent but the type (kind,set,etc) of referents, the `ctp` attribute is used.

```
You bought <np id="c1">a car</np>.
I bought <np ctp="c1">one</np>, too.
```

The values for the `rel` attribute also function as attributes, called *relational attributes*. A zero anaphora is encoded by a relational attribute.

```
Tom visited <np id="m1">Mary</np>.
He had <v iob="m1">brought</v> a
present.
```

`iob="m1"` means that the indirect object of *brought* is element `m1`, that is, *Mary*.

Other relational attributes in this connection include `sub` and `sup`. `sub` represents subset, part, or element. An example is:

```
She has <np id="b1">many books</np>.
<namep sub="b1">''Alice's Adventures
in Wonderland''</namep> is her
favorite.
```

`sup` is the inverse of `sub`, i.e., includer of any sort, which is superset as to subset, whole as to part, or set as to element.

3 Making Slide Show

We have developed a system which generates slide shows from GDA-tagged documents. Our method for slide presentation consists of two aspects. The first is to detect topics in the given document. The second aspect is to generate slides for the topics and organize them to a slide show. The latter employs some language-dependent heuristics. But neither aspect uses any heuristics dependent on the domain and/or style of documents. So our method is potentially applicable to any GDA-tagged documents.

3.1 Topic Detection

Topics are often represented by important words and/or phrases in the documents. A traditional method for topic identification is to use word/phrase-occurrence frequencies to extract such expressions. Such a method is not adequate for extracting topics, however, due to the following reasons:

1. A word is often too short to fully reresent a topic.
2. A topic is often represented by a variety of expressions.

For example, if we count the frequencies of the words in an article of the Wall Street Journal, which is in Figure 2, discard the words whose frequencies are less than two, and drop stop words, then we get

> PCs(6), Apple(5), PC(3), data(3), computers(3), years(2), year(2), market(2), IBM(2), times(2), Gates(2), business(2),

where the numbers are the frequencies. From this list, we know the article is about PCs. But it is doubtful that the list distinguishes the article from other articles which also describe PCs.

To remedy these problems, we may extract word bigrams in addition to word unigrams or use a stemmer to normalize expressions. But these are not fundamental solutions.

Instead we use semantic dependencies and coreferences for identifying topics. First we collect syntactic subjects and classify them according to their referents, and then discard the classes consisting of less than two elements. Next, we choose representative expressions from these classes and regard them as topics. A representative expression of a class is the element which is assigned the `id` attribute related with the class unless the element is elaborated by another element. If it is elaborated, then the elaborating expression is selected as representative.

For example, we can extract the following four topics from the WSJ article.

- the Apple II , Commodore Pet and Tandy TRS (5)
- Apple II (2)
- many pioneer PC contributors (4)
- IBM (2)

where the numbers are the sizes of the classes. Note that "the Apple II , Commodore Pet and Tandy TRS" does not have an `id` attribute because it is a coreference expression whose antecedent is "THREE COMPUTERS THAT CHANGED the face of personal computing." Nevertheless it is selected as a topic because it elaborates its antecedent. Note also that "many pioneer PC contributors" is not a subject but it is selected as the representative expression of "William Gates and Paul Allen," "Gates," "Alan F. Shugart," and "Dennis Hayes and Dale Heatherington" because it has an `id` attribute and is pointed by the other expressions with `sub` relation.

We believe that the expressions extracted by using syntactic and coreference information is much more appropriate for topics than the ones based on word frequencies. It is, however, a future work to confirm it experimentally.

Topic Selection

Frequency is not enough to distinguish the importances of topics (words and/or phrases) because different topics often have the same frequency. So we use a sort of spreading activation [Nagao and Hasida, 1998] to calculate the importance of elements. A GDA-tagged document is regarded as a network in which nodes correspond to GDA elements and links represent the syntactic

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 world-wide.

Figure 2: An article of the Wall Street Journal

dominance and semantic relationships described before. That is, this network is the tree of GDA elements plus cross-reference links among the nodes therein. Spreading activation applies to this network. It is performed respecting the condition that two elements should have the same activation value if either they are coreferent or one of them is a syntactic head of the other.

When we apply spreading activation to the WSJ article, we get the following activation values for the topics:

- (9.61) the Apple II , Commodore Pet and Tandy TRS
- (7.29) many pioneer PC contributors
- (4.67) IBM
- (4.65) Apple II

We can pick up the top two as the most important topics which will be presented in the slide show if we discard the topics whose activation values are smaller than a half of that of the top topic. We can also display this whole list to the audience so that he/she/they can choose topics to be presented in the rest of the slide show.

3.2 Slide Generation

A slide show is created by composing a slide for each topic selected as discussed above. In the current implementation of the slide presentation system, each slide is basically an itemized summary of the segment concerning the topic.

The initial slide may be a table of contents of the whole slide show, which is compiled by listing the topics. Each slide in the main body of the presentation is composed by following the steps below. Here a *topical element* is an GDA element linked with the topic via the `eq`, `ctp`, `sub`, or `sup` relation. A topical element which is the subject of a whole sentence is called a *topical subject*.

1. Let the topic be the heading of the slide.
2. Extract important sentences which contain topical subjects.
3. Remove redundant sentences, such as one elaborated by another extracted sentence, where elaboration is encoded by the `ela` relation.
4. Itemize the remaining sentences by the following heuristics, among many others.
   (a) Prune unimportant expressions such as some (typically unrestrictive) relative clauses and appositive phrases.
   (b) Remove the topical subjects linked with the topic through the `eq` or `ctp` relation.
   (c) Pronominalize non-subject topical elements linked with the topic through the `eq` or `ctp` relation.
   (d) Emphasize the topical elements linked with the topic through the `sub` or `sup` relation.
   (e) Replace non-topical anaphoric elements with their antecedents.
   (f) Move the elements preceding the removed topical subjects to the end of the sentences.
   (g) Decompose coordinate structures whose conjunctions are *and*, *as well as*, *not only* $\sim$ *but also*, etc. into separate items.

Heuristics (a) through (g) are specific to English, but it is straightforward to adapt them to other languages. The above WSJ article eventually gives rise to the three slides in Figure 3, Figure 4, and Figure 5.

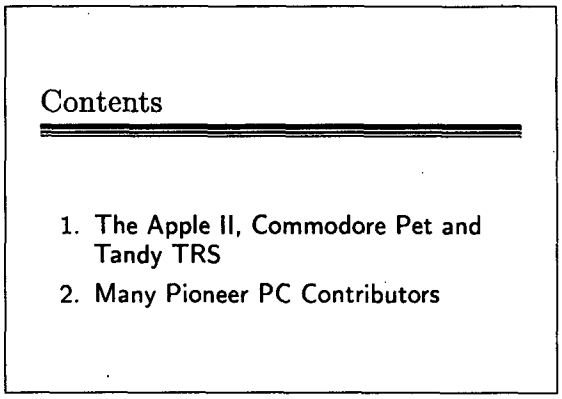

Figure 3: The first slide.

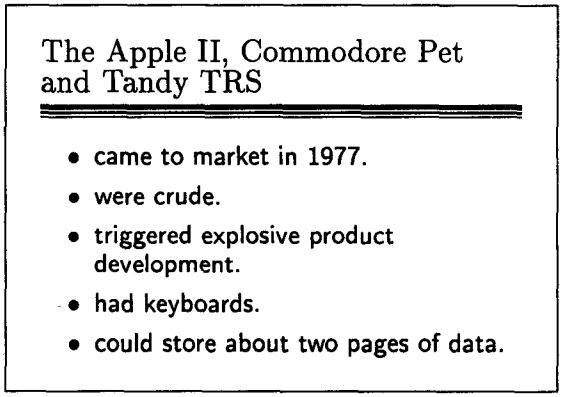

Figure 4: The second slide.

The first slide in Figure 3 is the table of contents. The second slide is titled by the first topic in the article, followed by a list of items. To compose this list, initially the following sentences are picked up which talk about the topic.

1. THREE COMPUTERS THAT CHANGED the face of personal computing were launched in 1977.
2. That year the Apple II, Commodore Pet and Tandy TRS came to market.
3. The computers were crude by today's standards.
4. Crude as they were, these early PCs triggered explosive product development in desktop models for the home and office.

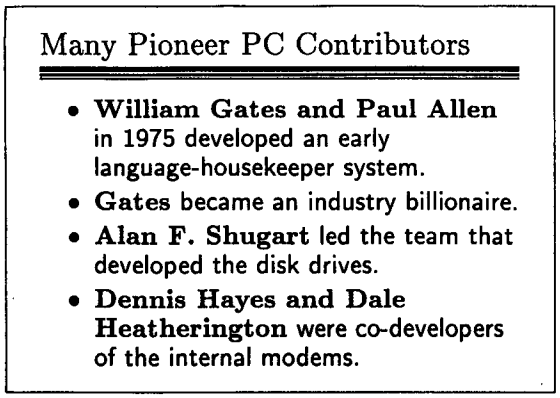

Figure 5: The third slide.

5. 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.

The first sentence is abandoned because it is elaborated by the second. In the other sentences, unnecessary subexpressions are pruned off due to (a) and the references to the topic are replaced by ϕ due to (b), as follows:

1. That year ϕ came to market.
2. ϕ were crude.
3. ϕ triggered explosive product development.
4. ϕ had keyboards and could store about two pages of data.

The first sentence above is then paraphrased by replacing "that year" with "in 1977" due to (e) and moving it at the end due to (f). The coordinate structure in the last sentence is decomposed into two list items due to (g). The final result is the slide shown in Figure 4.

The third slide is composed in essentially the same way as the second, except that the topical subjects are emphasized due to (d) as shown in Figure 5. Further details are omitted.

From preliminary experiments, we found that the above heuristics work fine for many cases. But in some cases they break down. For example, applying heuristic (a) to "The Wall Street Journal will report events of the past century that stand as milestones of American business history." produces "The Wall Street Journal will report events," which is not appropriate because the resulting sentence lacks the information necessary to describe what event the WSJ is going to report. Such a problem may be avoided if there are pragmatic tags to encode which parts of the document somehow convey new information.

3.3 Dynamic Adaptation

Under the framework described so far, it is straightforward to dynamically adapt a presentation to the audience's requests. This is done by reflecting interactions with the audience in the evaluation of importance and topic selection. This adaptation of importance evaluation and topic selection leads to reorganization of the presentation.

The current presentation system deals with a simple type of interaction which allows the audience to issue questions about parts of the document. This is done in two ways, one by clicking on the screen and the other by typing on the keyboard. A click on a point in a slide is to select the smallest element containing that point. A further click on the selected element is to select its parent element, and so forth. Having specified a part of the document, whether by clicking or typing, the audience can then request an explanation about it. A new slide is made and shown on the fly if the original document contains more information (absent in the present slide) about that phrase. The remaining part of the presentation, if any, incorporates such interaction by evaluating the specified phrase more importance than otherwise.

For instance, suppose the audience asks about 'IBM' at some point in the slide show from Figure 3 to Figure 5. Then a slide shown in Figure 6 will be composed

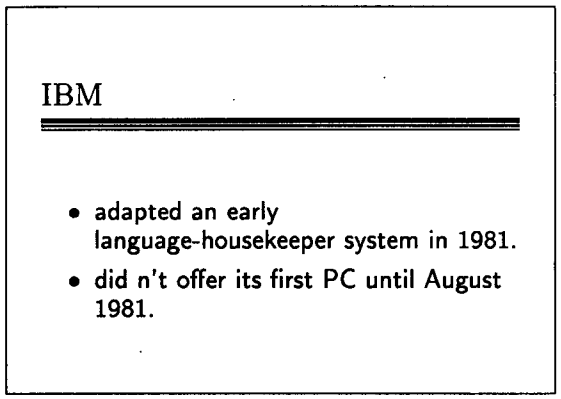

Figure 6: An improvised slide.

extempore.

4 Concluding Remarks

We have discussed automatic generation of slide presentations from semantically annotated documents. The reported presentation system first detects important topics in the given document and then creates a slide for each topic. Coreferences play a central role in both topic identification and paraphrasing summarization.

The presentation can be dynamically customized by reflecting the interaction with the audience in topic selection and importance evaluation. Since the GDA tagset is independent of the domain and style of documents and also applicable to diverse natural languages, the reported system is domain/style-free and easy to adapt to different languages as well.

There is no established formal method for evaluating a technology such as slide presentation. We are hence attempting to evaluate partial aspects of the reported

method, such as topic selection and paraphrasing. A more synthetic evaluation is a future work.

There are several avenues along which to improve or extend the reported system. First, it should be easy to incorporate figures and tables into the slides from the original document. These non-textual materials can also be treated as GDA elements and processed in the same way as text elements with respect to importance evaluation. Second, textual materials could often be rendered visually more perspicuous than a mere list of items. For instance, some sorts of textual content could be naturally depicted by a graph with labeled nodes and arrows, on the basis of spatial metaphors. Third, not just subjecthood but also other grammatical functions and anaphoricity of the relevant expressions could be used to identify topics. The intuitions behind centering theory [Grosz *et al.*, 1995] may be useful here. Finally, more sophisticated types of interaction than described above are desirable and feasible, including question answering.

References

[Grosz *et al.*, 1995] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Centering: A framework for modeling the local coherence of discource. *Computational Linguistics*, 21(2):203–225, 1995.

[Hasida, 1997] Kôiti Hasida. Global Document Annotation. In *Natural Language Processing Pacific Rim Symposium '97*, 1997.

[Marcus *et al.*, 1993] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[Nagao and Hasida, 1998] Katasi Nagao and Kôiti Hasida. Automatic text summarization based on the Global Document Annotation. In *COLING-ACL '98*, pages 917–921, 1998.

[Perkovitz and Etzioni, 1997] Mike Perkovitz and Oren Etzioni. Adaptive web sites: an AI challenge. In *IJCAI'97*, 1997.

[Perkovitz and Etzioni, 1998] Mike Perkovitz and Oren Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *AAAI'98*, 1998.