Semantic Annotation and Transcoding: Making Web Content More Accessible

Katashi Nagao
IBM Research

Yoshinari Shirai
NTT Communication Science Labs

Kevin Squire
University of Illinois

We propose a method for constructing a superstructure on the Web using XML and external annotations to Web documents. We have three approaches for annotating documents—linguistic, commentary, and multimedia. The result is annotated documents that computers can understand and process more easily, allowing content to reach a wider audience with minimal overhead.

Most of us have created Web content using only human-friendly information. Of course, this makes sense—after all, humans are the intended users of the information. However, with the continuing explosion of Web content—especially multimedia content—we argue that contents should be created in a more machine-friendly format—one that lets machines better understand and process documents. In this article, we propose a system to annotate documents externally with more information in order to make them easier for computers to process. Hori et al.[1] suggested the original concept of external annotation. Here we use this term in a more general sense.

Why would we want to make content more machine friendly? Well, annotated documents are easier to personalize. For example, a computer can process textual information annotated with parts of speech and word senses much better than plain text and can therefore produce a nice, grammatically correct personalized summary formatted for a cellular phone—or translate a document from English to Japanese. Normally, when dealing with nonannotated text, transcoding to many different formats requires a lot of task-specific effort for each format. However, by using document annotation, content providers put in some extra work early on, but receive the benefits of being able to transcode to an endless variety of formats and personal tastes easily, thus reaching a much wider audience with less overall effort.

Overview

Our idea of a Web superstructure consists of layers of content and metacontent. The first layer corresponds to the set of metacontents of base documents. The second layer corresponds to the set of metacontents of the first layer, and so on. We generally consider such metacontent as external annotation. A good example of external annotation is external links that can be defined outside of the set of link-connected documents. We've discussed these external links in the Extensible Markup Language (XML) community but they haven't been implemented yet in the current Web architecture (see http://www.w3.org/XML/).

Another popular example of external annotation is comments or notes on Web documents created by people other than the author. This kind of annotation is useful for readers evaluating the documents. For example, images without alternative descriptions aren't understandable for visually challenged people. If comments accompany these images, these people can understand the image contents by listening via speech transcoding.

We can easily imagine that an open platform for creating and sharing annotations would greatly extend the Web's expressive power and value. Many of the benefits our system offers could be obtained by converting the underlying content into suitably annotated documents. While this conversion may eventually take place on the Web, our system allows easy adaptation of existing Web documents for transcoding, even when the annotator isn't the owner of the original document.

Recent activities on the Semantic Web (http://www.semanticweb.org) try to annotate Web documents with ontology-based semantic descriptions based on the Resource Description Framework, or RDF (see http://www.w3.org/TR/REC-rdf-syntax/). Their approach seems topdown and there's a big gap between the current Web and the Semantic Web. Our approach is bottom-up and our annotation system lets ordinary people annotate their Web documents with some additional and useful information by using our user-friendly tools.

Content adaptation

Annotations don't just increase the expressive power of the Web—they also play an important role in content reuse. An example of content reuse is the transformation of content depending on user preferences.

Content adaptation is a type of transcoding that considers a user's environment—devices, network bandwidth, profiles, and so on. In addition, such adaptation sometimes requires a good understanding of the original document contents. If the transcoder fails to analyze a document's semantic structure, then the transcoding results may not be accurate and may cause user misunderstanding.

Our technology assumes that external annotations help machines understand document contents so that transcoding can have higher quality. We call such transcoding based on annotation semantic transcoding. Figure 1 shows the overall configuration of the semantic transcoding system.

Our system features three new parts: an annotation editor, an annotation server, and a transcoding proxy server. The remaining parts of the system consist of a conventional Web server and a browser.

IBM published previous work on device-dependent adaptation of Web documents at http://www-4.ibm.com/software/webservers/transcoding/. The system it developed can dynamically filter, convert, or reformat data for content sharing across disparate systems, users, and pervasive computing devices.

IBM claims that the transcoding benefits include

1. eliminating the expense of reauthoring or porting data and content-generating applications for multiple systems and devices,

2. improving mobile employee communications and effectiveness, and

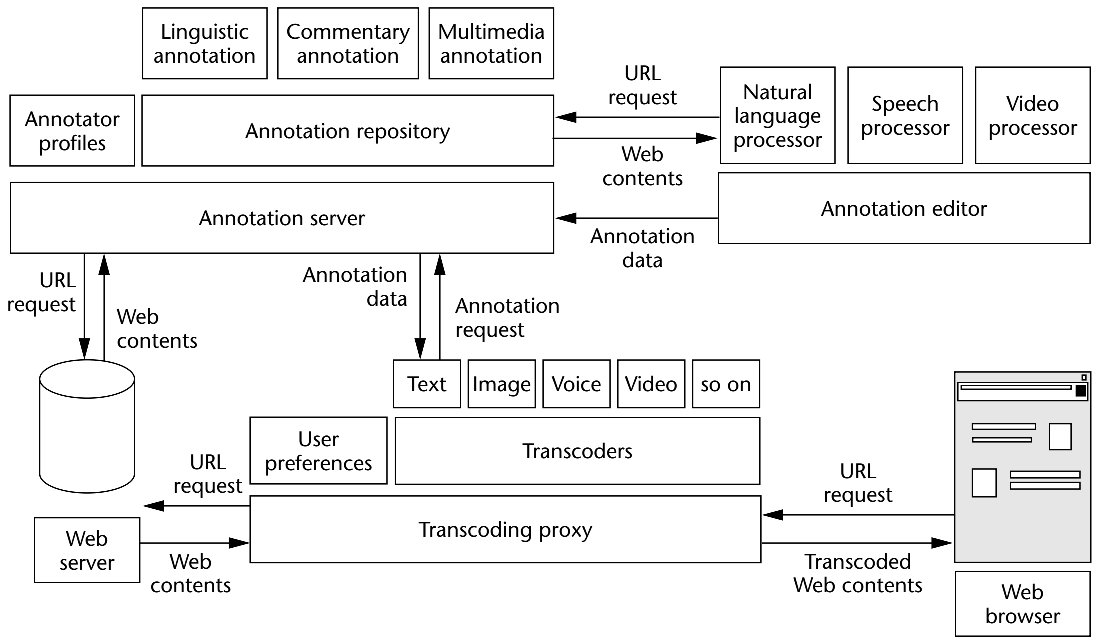
3. creating easier access for customers who use a variety of devices to purchase products and services.

The technology enables the modification of Web documents, such as converting images to links to retrieve images, converting simple tables to bulleted lists, removing features not supported by a device such as JavaScript or Java applets, removing references to image types not supported by a device, and removing comments. It can also transform XML documents by selecting and applying the right stylesheet for the current request based on information in the relevant profiles. These profiles for preferred transcoding services are defined for an initial set of devices.

Our transcoding system involves deeper levels of document understanding. Therefore, it needs human intervention into machine understanding of documents. External annotation is a guide for machine understanding. Of course, some profiles for user contexts will work as a guide for transcoding, but it's clear that such profiles are insufficient for transcoders to recognize and manipulate fundamental document features.

Various researchers

Figure 1. Semantic transcoding system.



have studied content personalization for some specific purposes. Brusilovsky et al.[2] developed an online textbook authoring tool that adapts to students' level of understanding. Their technique involves embedding tags into the original document. Milosavljevic et al.[3] developed systems for dynamically producing documents from abstract data and information about the user. In both cases, the amount and type of personalization or adaptation is limited compared to our system. It's also much more difficult to adapt current contents to fit these systems, whereas our system will work immediately with content anywhere on the Web.

Fink et al.[4] discussed a technique on adapting Web information for handicapped individuals. We're also planning to extend our transcoding work for accessibility, which means all people—including seniors and any kind of handicapped people—can easily use information systems.

Knowledge discovery

Another use of annotations is in knowledge discovery, where intelligent software agents mine huge amounts of Web contents automatically for some essential points. Unlike conventional search engines that retrieve Web pages using user-specified keywords, knowledge miners create a single document that satisfies a user's request. For example, the knowledge miner may generate a summary document on a certain company's product strategy for the year from many kinds of information resources of its products on the Web.

Currently, we're developing an information collector that gathers documents related to a topic and generates a document containing a summary of each document.

External annotation

We developed a simple method for associating external annotations with any element of any HTML document. We use URLs, XPaths, and document hash codes (digest values) to identify HTML elements in documents. We also developed an annotation server that maintains the relationship between contents and annotations and transfers requested annotations to a transcoder.

We represent our annotations as XML formatted data and divide them into three categories—linguistic, commentary, and multimedia annotation. Multimedia annotation (especially of video) uses a combination of the other two types of annotation.

Annotation environment

Our annotation environment consists of a client-side editor for the creation of annotations and a server for the management of annotations. Figure 2 shows the annotation environment.

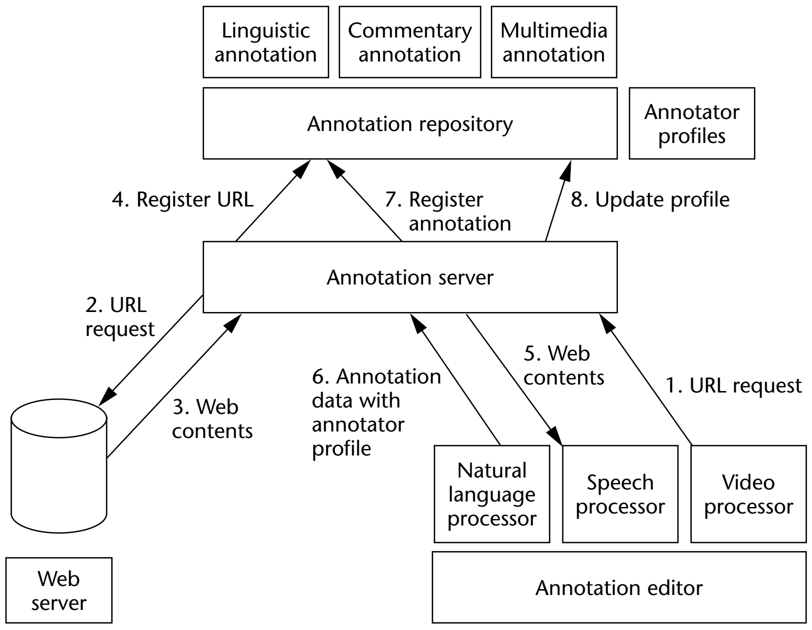
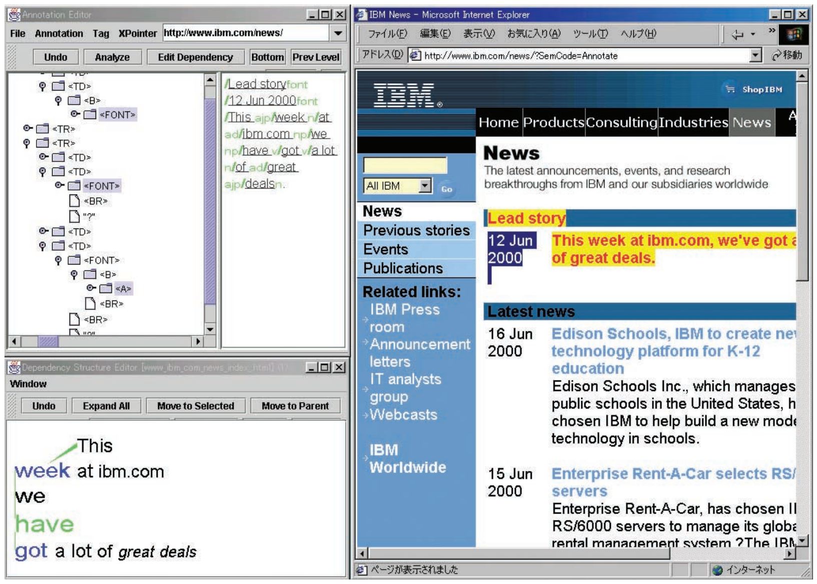


Figure 2. Annotation environment.

The process flows as follows (in this example, the server processes HTML files):

1. The user runs the annotation editor and requests an URL as a target of annotation.
2. The annotation server accepts the request and sends it to the Web server.
3. The annotation server receives the Web document.
4. The server calculates the document hash code (digest value) and registers the URL with the code to its database.
5. The server returns the Web document to the editor.
6. The user annotates the requested document and sends the result to the server, along with some personal data (name, professional areas, and so on).
7. The server receives the annotation data and relates it to its URL in the database.
8. The server also updates the annotator profiles.


Annotation Editor
File Annotation Tag XPointer http://www.ibm.com/news/
Undo Analyze Edit Dependency Bottom Prev Level
Dependency Structure Editor
Window
Undo Expand All Move to Selected Move to Parent
IBM News - Microsoft Internet Explorer
Home Products Consulting Industries News
Shop IBM

/Lead story font /12 Jun 2000 font /This ajp /week n /at ad /ibm.com np /we np /have v /got v /a lot n /of ad /great ajp /deals n.

This week at ibm.com we have got a lot of great deals

News
The latest announcements, events, and research breakthroughs from IBM and our subsidiaries worldwide

Lead story
12 Jun 2000 This week at ibm.com, we've got a lot of great deals.

Latest news
16 Jun 2000 Edison Schools, IBM to create new technology platform for K-12 education
Edison Schools Inc., which manages public schools in the United States, has chosen IBM to help build a new model for technology in schools.
15 Jun 2000 Enterprise Rent-A-Car selects RS/6000 servers
Enterprise Rent-A-Car, has chosen IBM RS/6000 servers to manage its global rental management system.

News
Previous stories
Events
Publications
Related links:
IBM Press room
Announcement letters
IT analysts group
Webcasts
IBM Worldwide

Figure 3. Screen shot of the annotation editor.

Annotation editor

Our annotation editor, implemented as a Java application, can communicate with the annotation server (explained in the next section).

The annotation editor

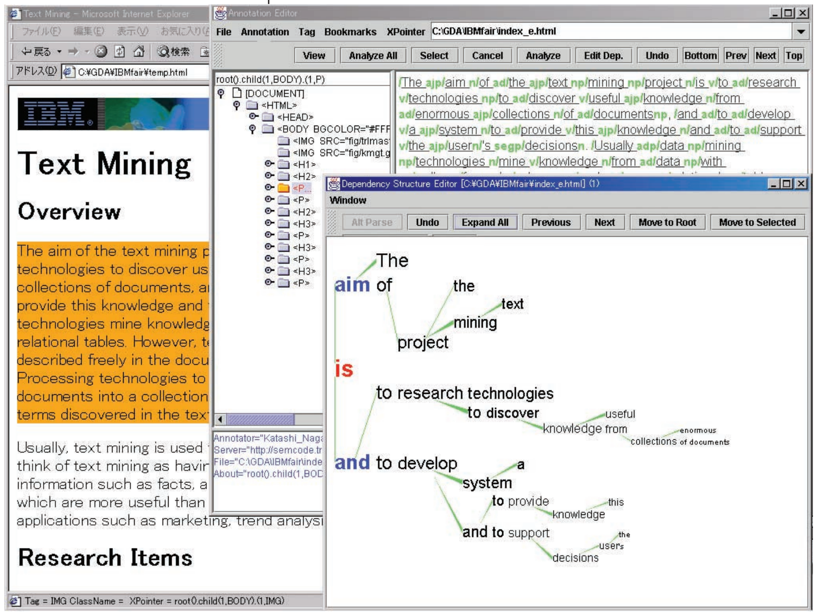
1. registers targets of annotation with the annotation server by sending URLs,

2. specifies elements in the document using a Web browser,

3. generates and sends annotation data to the annotation server, and

4. reuses previously created annotations when the target contents are updated.

Figure 4. Screen shot of the annotation editor with linguistic structure editor.


Text Mining - Microsoft Internet Explorer
Annotation Editor
File Annotation Tag Bookmarks XPointer C:\GDA\IBMfair\index_e.html
View Analyze All Select Cancel Analyze Edit Dep. Undo Bottom Prev Next Top
Dependency Structure Editor [C:\GDA\IBMfair\index_e.html] (1)
Window
Alt Parse Undo Expand All Previous Next Move to Root Move to Selected
Tag = IMG ClassName = XPointer = root0.child(1,BODY).(1,IMG)

/The ajp /aim n /of ad /the ajp /text np /mining np /project n /is v /to ad /research v /technologies np /to ad /discover v /useful ajp /knowledge n /from ad /enormous ajp /collections n /of ad /documents np. /and ad /to ad /develop v /a ajp /system n /to ad /provide v /this ajp /knowledge n /and ad /to ad /support v /the ajp /users n /'s segp /decisions n. /Usually adp /data np /mining np /technologies n /mine v /knowledge n /from ad /data np /with

The aim of the text mining project is to research technologies to discover useful knowledge from enormous collections of documents and to develop a system to provide this knowledge and to support the users decisions

Text Mining
Overview
The aim of the text mining project is to research technologies to discover useful knowledge from enormous collections of documents, and to provide this knowledge and to support the users' decisions. Usually, data mining technologies mine knowledge from relational tables. However, text is described freely in the documents. Processing technologies to convert documents into a collection of terms discovered in the text
Usually, text mining is used to think of text mining as having information such as facts, and which are more useful than applications such as marketing, trend analysis
Research Items

Annotator="Katashi_Nagao" Server="http://semcode.tr" File="C:\GDA\IBMfair\index_e.html" About="root0.child(1,BODY).(1,P)"

Figure 3 shows an example screen of our annotation editor. The left top window of the editor shows the HTML's document object structure. The right window shows some text selected on the Web browser. The editor automatically assigns the selected area an XPath—for example, a location identifier in the document (see http://www.w3.org/TR/xpath.html). The left bottom window shows the linguistic structure of the sentence in the selected area.

Using the editor, the user annotates text with the linguistic structure (grammatical and semantic structure, described later) and adds a comment to an element in the document. The editor is capable of basic natural language processing and interactive disambiguation. The user should verify and modify the results of the automatically analyzed sentence structure (Figure 4).

Annotation server

Our annotation server receives annotation data from an annotator and classifies it according to the annotator's name. The server retrieves documents from URLs in annotation data and registers the document hash codes with their URLs in its annotation database. Since a document's author may modify the document after the initial retrieval, the hash code of a document's internal structure or DOM (Document Object Model) enables the server to discover modified elements in the annotated document.[5]

The annotation server makes a table of annotator names, URLs, XPaths, and document hash codes. When the server accepts a URL as a request from a transcoding proxy, the server returns a list of XPaths with associated annotation files, their types (linguistic or commentary), and a hash code. If the server receives an annotator's name as a request, it responds with the set of annotations that the specified annotator creates.

We're currently developing a mechanism for access control between annotation servers and normal Web servers. If authors of original documents don't want anyone to annotate their documents, they can add a statement about it in the documents, and annotation servers won't retrieve such contents for the annotation editors.

Linguistic annotation

The purpose of linguistic annotation is to make text on the Web machine understandable (on the basis of a new tag set) and to develop better quality content-based presentation, retrieval, question–answering, summarization, and translation systems. The Global Document Annotation (GDA) project proposed a new tag set (see http://www.i-content.org/GDA/). It's based on XML and is designed to be as compatible as possible with HTML, the Text Encoding Initiative (TEI, see http://www.uic.edu:80/orgs/tei), Corpus Encoding Standard (CES, http://www.cs.vassar.edu/CES/), Expert Advisory Group on Language Engineering, (Eagles, http://www.ilc.pi.cnr.it/EAGLES/home.html), and Linguistic Annotation Language (LAL).[6] It specifies modifier–modifiee relations, anaphor-referent relations, word senses, and so on.

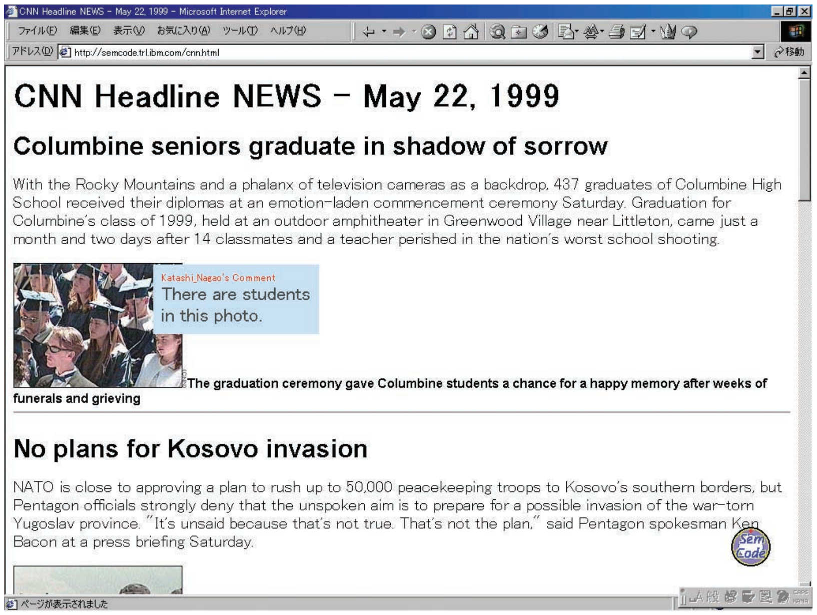
A GDA-tagged sentence looks like

```
<su><np rel="agt" sense="time0">
     Time</np>
<v sense="fly1">flies</v>
<adp rel="eg">
     <ad sense="like0">like</ad>
<np>an <n sense="arrow0">arrow</n>
     </np>
</adp>.</su>
```

The tag `<su>` refers to a sentential unit. The other tags above—`<n>`, `<np>`, `<v>`, `<ad>`, and `<adp>`—mean noun, noun phrase, verb, adnoun or adverb (including preposition and postposition), and adnominal or adverbial phrase, respectively. A more detailed description of the GDA tag set can be found at http://www.i-content.org/GDA/tagset.html.

The `rel` attribute encodes a relationship in which the current element stands with respect to the element on which it semantically depends. Its value is called a relational term. A relational term denotes a binary relation—which may be a thematic role such as agent, patient, or recipient—or a rhetorical relation such as cause, concession, and so on. For instance, in the GDA-tagged sentence, `<np rel="agt" sense="time0">Time</np>` depends on the second element `<v sense="fly1">flies</v> rel="agt"`, which means that `Time` has the agent role with respect to the event denoted by `flies`. Finally, the `sense` attribute encodes a word sense.

Automatic morphological analysis, interactive sentence parsing, and word sense disambiguation generate linguistic annotation by selecting the most appropriate paraphrase. Some research issues on linguistic annotation relate to reducing annotation cost within some feasible levels. We've been developing some machine-guided annotation interfaces that conceal the complexity of annotation. Machine-learning mechanisms also contribute to reducing the cost because they can gradually increase the automatic annotation accuracy.


CNN Headline NEWS – May 22, 1999 – Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
アドレス(D) http://semcode.trl.ibm.com/cnn.html
移動
ページが表示されました
Sem Code

CNN Headline NEWS – May 22, 1999

Columbine seniors graduate in shadow of sorrow

With the Rocky Mountains and a phalanx of television cameras as a backdrop, 437 graduates of Columbine High School received their diplomas at an emotion-laden commencement ceremony Saturday. Graduation for Columbine's class of 1999, held at an outdoor amphitheater in Greenwood Village near Littleton, came just a month and two days after 14 classmates and a teacher perished in the nation's worst school shooting.

Katashi_Nagao's Comment
There are students in this photo.

The graduation ceremony gave Columbine students a chance for a happy memory after weeks of funerals and grieving

No plans for Kosovo invasion

NATO is close to approving a plan to rush up to 50,000 peacekeeping troops to Kosovo's southern borders, but Pentagon officials strongly deny that the unspoken aim is to prepare for a possible invasion of the war-torn Yugoslav province. "It's unsaid because that's not true. That's not the plan," said Pentagon spokesman Ken Bacon at a press briefing Saturday.

Figure 5. Comment overlay on the document.

In principle, the tag set doesn't depend on language, but as a first step we implemented a semi-automatic tagging system for English and Japanese.

Commentary annotation

Commentary annotation mainly annotates nontextual elements like images and sounds with some additional information. Each comment can include tagged texts and other images and links. Currently, this type of annotation appears in a subwindow overlayed on the original document window when a user locates a mouse pointer at the area of a comment-added element (Figure 5).

Users can also annotate text elements with information such as paraphrases, correctly spelled words, and underlines. We use this type of annotation for text transcoding that combines such comments on texts and original texts.

Our system also features commentary annotation on hyperlinks. This contributes to quick introduction of target documents before clicking on the links. If there are linguistic annotations on the target documents, the transcoders can generate summaries of these documents and relate them to hyperlinks in the source document.

Previously, some research has been published concerning sharing comments over the Web. ComMentor is a general metainformation architecture for annotating documents on the Web.[7] This architecture includes a basic client-server protocol, metainformation description language, server system, and remodeled NCSA Mosaic browser with interface augmentations to provide access to its extended functionality. ComMentor provides a general mechanism for shared annotations, which lets people annotate arbitrary documents at any position in-place, share comments or pointers with other people (either publicly or privately), and create shared landmark reference points in the information space. Several annotation systems have a similar direction, such as the group annotation transducer.[8]

These systems are often limited to particular documents or documents shared only among a few people. Our annotation and transcoding system can handle multiple comments on any element of any document on the Web. Also, we can add a community-wide access control mechanism to our transcoding proxy. If a user isn't a member of a particular group, then the user can't access the transcoding proxy that's for group use only. In the future, transcoding proxies and annotation servers will communicate with some secure protocol that prevents some other server or proxy from accessing the annotation data.

Our main focus is the adaptation of Web contents to users, and sharing comments in a community is one of our additional features. We apply both commentary and linguistic annotations to semantic transcoding.

Multimedia annotation

We can also apply our annotation technique to multimedia data, such as digital video. Digital video is becoming a prevalent information source. Since these collections are growing to a huge number of hours, it's necessary to summarize and browse in a short time without losing significant content. We've developed techniques for semiautomatic video annotation use text to describe a video's content. Our techniques use some video analysis methods, such as automatic cut detection, characterization of frames in a cut, and scene recognition using similarity between several cuts.

Other approaches to video annotation exist. For example, MPEG-7 is an effort within the Moving Picture Experts Group (MPEG) of the International Organization for Standardization/International Electronics Commission (ISO/IEC) that's dealing with multimedia content description (see http://www.drogo.cselt.stet.it/mpeg/standards/mpeg-7/mpeg-7.htm).

Using content descriptions, video coded in MPEG-7 is concerned with transcoding and delivery of multimedia content to different devices. MPEG-7 will potentially allow greater input from content publishers in guiding how to transcode multimedia content in different situations and for different client devices. Also, MPEG-7 provides object-level descriptions of multimedia content. This allows a higher granularity of transcoding where individual regions, segments, objects, and events in image, audio, and video data can transcode differentially—depending on publisher and user preferences, network bandwidth, and client capabilities.

Our method will be integrated into tools for authoring MPEG-7 data. However, we don't currently know when the MPEG-7 technology will be widely available.

Our video annotation includes automatic segmentation of video, semiautomatic linking of video segments with corresponding text segments, and interactive naming of people and objects in video frames.

Video annotation occurs through the following steps:

1. For each video clip, the annotation system creates the text corresponding to its content. We employ speech recognition for the automatic generation of a video transcript. The speech recognition module also records correspondences between the video frames and the words. The transcript isn't required to describe the whole video content. The resolution of the description affects the final quality of the transcoding (for example, summarization).

2. The software applies some video analysis techniques to characterize scenes, segments (cuts and shots), and individual frames in video. For example, by detecting significant changes in the color histogram of successive frames, the application can separate frame sequences into cuts and shots.

   Also, by searching and matching prepared templates to individual regions in the frame, the annotation system identifies objects. The user can specify significant objects in a scene to reduce the time for identifying target objects and to obtain a higher recognition success ratio. The user can name objects in a frame by selecting words in the corresponding text.

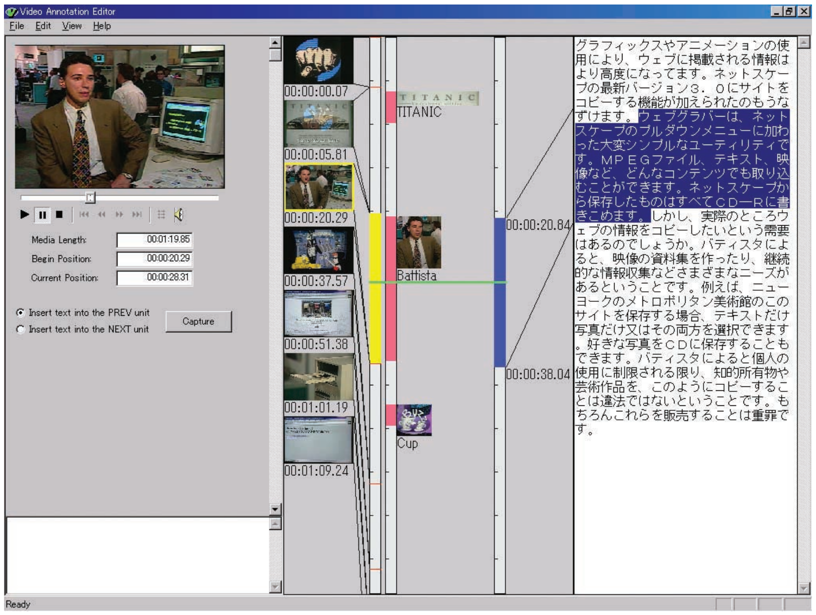
3. The user relates video segments to text seg-

ments such as paragraphs, sentences, and phrases, based on scene structures and object-name correspondences. The system helps the user select appropriate segments by prioritizing based on the number of objects detected, camera movement, and by showing a representative frame of each segment.

We developed a video annotation editor capable of scene change detection, object tracking, speech recognition, and correlation of scenes and words. Figure 6 shows a screen shot of our video annotation editor.

On the editor screen, the user can specify a particular object in a frame by dragging a rectangle. Using automatic object-tracking techniques, the annotation editor can generate descriptions of an object in a video frame. XML data represents the description and contains object coordinates in start and end frames, time codes of the start and end frames, and motion trails (series of coordinates for the interpolation of object movement).

The object descriptions connect with linguistic annotation by adding appropriate XPaths to the tags of corresponding names and expressions in the video transcript.



Media Length: 00:01:19.85
Begin Position: 00:00:20.29
Current Position: 00:00:28.31

[✓ Insert text into the PREV unit]
Insert text into the NEXT unit

グラフィックスやアニメーションの使用により、ウェブに掲載される情報はより高度になってます。ネットスケープの最新バージョン３．０にサイトをコピーする機能が加えられたのもうなずけます。ウェブグラバーは、ネットスケープのプルダウンメニューに加わった大変シンプルなユーティリティです。ＭＰＥＧファイル、テキスト、映像など、どんなコンテンツでも取り込むことができます。ネットスケープから保存したものはすべてＣＤ−Ｒに書きこめます。しかし、実際のところウェブの情報をコピーしたいという需要はあるのでしょうか。バティスタによると、映像の資料集を作ったり、継続的な情報収集などさまざまなニーズがあるということです。例えば、ニューヨークのメトロポリタン美術館のこのサイトを保存する場合、テキストだけ写真だけ又はその両方を選択できます。好きな写真をＣＤに保存することもできます。バティスタによると個人の使用に制限される限り、知的所有物や芸術作品を、このようにコピーすることは違法ではないということです。もちろんこれらを販売することは重罪です。

Figure 6. Screen shot of the video annotation editor.

Semantic transcoding

Semantic transcoding is a transcoding technique based on external annotations, used for content adaptation according to user preferences. The transcoders here are implemented as an extension to an HTTP proxy server. Such an HTTP proxy is called a transcoding proxy. Figure 7 shows the environment of semantic transcoding.

The information flow in transcoding occurs as follows:

1. The transcoding proxy receives a request URL with a client ID.

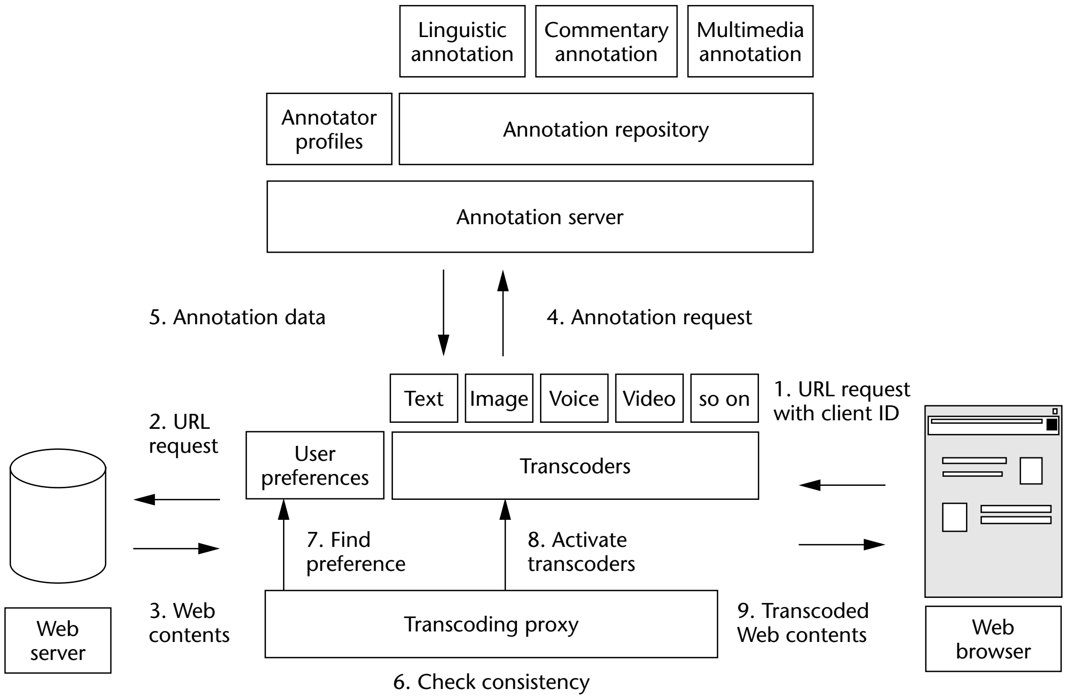


Figure 7. Transcoding environment.

2. The proxy sends the URL's request to the Web server.

3. The proxy receives the document and calculates its hash code.

4. The proxy also asks the annotation server for annotation data related to the URL.

5. If the server finds the URL's annotation data in its database, it returns the data to the proxy.

6. The proxy accepts the data and compares the document hash code with that of the already retrieved document.

7. The proxy also searches for the user preference with the client ID. If no preference data exist, the proxy uses a default setting until the user gives the preference.

8. If the hash codes match, the proxy attempts to transcode the document based on the annotation data by activating the appropriate transcoders.

9. The proxy returns the transcoded document to the client Web browser.

Transcoding proxy

We employed IBM's Web intermediaries (WBI) as a development platform to implement the semantic transcoding system (see http://www.almaden.ibm.com/cs/wbi). WBI is a customizable and extendable HTTP proxy server. WBI provides application programming interfaces (APIs) for user-level access control and easy manipulation of input/output data of the proxy.

The transcoding proxy based on WBI has the following functionality:

1. maintenance of personal preferences,

2. gathering and management of annotation data,

3. activation and integration of transcoders.

User preference management. For the maintenance of personal preferences, we use the Web browser's cookie to identify the user. The cookie holds a user ID assigned by the transcoding proxy on the first access and uses the ID to identify the user and select user preferences previously defined. Storing the ID as a cookie value lets the user change an access point using the dynamic host configuration protocol (DHCP) with the same preference setting. However, there's one technical problem. Generally, only the HTTP servers that have their values set can access cookies. Ordinary proxies don't use cookies for user identification. Instead, conventional proxies identify the client by the hostname and IP address. Thus, when the user accesses our proxy and sets or updates the preferences, the proxy server acts as an HTTP server to access the browser's cookie data and associates the user ID (cookie value) to the hostname or IP address. When the transcoding proxy works as a coventional proxy, it receives the client's hostname and IP address, retrieves the user ID, and then obtains the preference data. If the user changes the access point and hostname or IP address, our proxy performs as a server again and reassociates the user ID to such client IDs.

Collecting and indexing annotation data. The transcoding proxy communicates with annotation servers that hold the annotation database. The second step of semantic transcoding is to collect annotations distributed among several servers.

The transcoding proxy creates a multiserver annotation catalog by traversing distributed annotation servers and gathering their annotation indexes. The annotation catalog consists of server name (for example, hostname and IP address) and its annotation index (set of annotator names and identifiers of the original document and its annotation data). The proxy uses the catalog to decide which annotation server to access to get annotation data when it receives a user's request.

Integrating the results of multiple transcoders. The final stage of semantic transcoding is to transcode requested contents depending on user preferences and then to return them to the user's browser. This stage involves activating appropriate transcoders and integrating their results.

As mentioned previously, several types of transcoding exist. In this article, we describe four types—text, image, voice, and video.

Text transcoding

Text transcoding is the transformation of text contents based on linguistic annotations. As a first step, we implemented text summarization.

Our text summarization method employs a spreading activation technique to calculate the importance values of elements in the text.[9] Since the method doesn't employ any heuristics dependent on the domain and style of documents, it's applicable to any linguistically annotated document. The method can also trim sentences in the summary because it assigns importance scores to elements smaller than sentences.

A linguistically annotated document naturally defines an intradocument network where nodes correspond to elements and links represent the semantic relations. This network consists of sentence trees (syntactic head–daughter hierarchies of subsentential elements such as words or phrases); coreference or anaphora links; document, subdivision, or paragraph nodes; and rhetorical relation links. Figure 8 shows the intradocument network.

The summarization algorithm works as follows:

1. Spreading activation performs in such a way that two elements have the same activation value if they're coreferent or one is the syntactic head of the other.

2. The algorithm marks the unmarked element with the highest activation value for inclusion in the summary.

3. When the algorithm marks an element, it recursively marks the following elements as well, until it can't find any more elements:

   - the marker's head

   - the marker's antecedent

   - the marker's compulsory or a priori important daughters, the values whose relational attributes are **agt** (agent), **pat** (patient), **rec** (recipient), **sbj** (syntactic subject), **obj** (syntactic object), **pos** (possessor), **cnt** (content), **cau** (cause), **cnd** (condition), **sbm** (subject matter), and so on

   - the antecedent of a zero anaphor in the marker with some of the above values for the relational attribute

4. The algorithm generates all marked elements in the intradocument network 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.

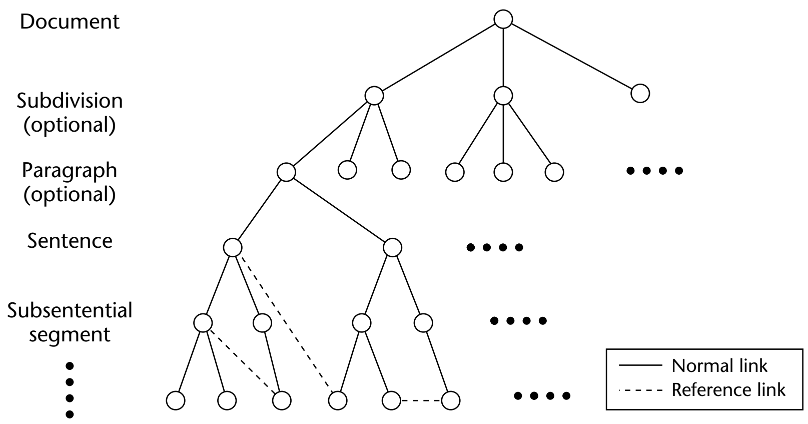
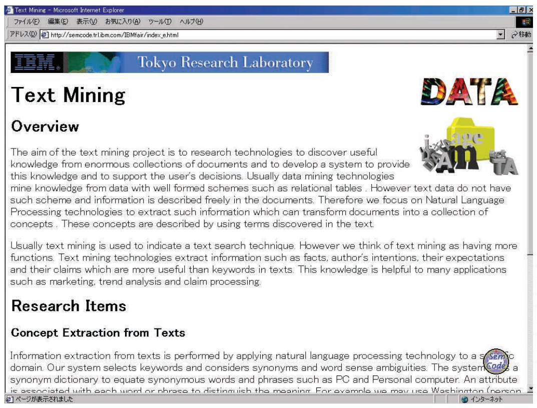


Figure 8. Intradocument network.

Simple user interaction can change the summary's size. Thus, the user can see the summary in a preferred size by using an ordinary Web browser without any additional software. The user can also input any words of interest. The algorithm assigns corresponding words in the document numeric values that reflect degrees of interest. The algorithm uses these values while spreading activation for calculating importance scores.

Figure 9 shows the summarization result on the normal Web browser. The top document is the orig-



Text Mining - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
アドレス(D) http://semcode.trl.ibm.com/IBMfair/index_e.html
移動
IBM Tokyo Research Laboratory
DATA
ページが表示されました
インターネット

Text Mining

Overview

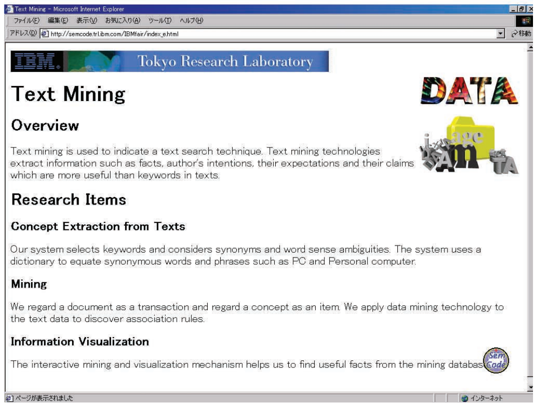
The aim of the text mining project is to research technologies to discover useful knowledge from enormous collections of documents and to develop a system to provide this knowledge and to support the user's decisions. Usually data mining technologies mine knowledge from data with well formed schemes such as relational tables . However text data do not have such scheme and information is described freely in the documents. Therefore we focus on Natural Language Processing technologies to extract such information which can transform documents into a collection of concepts . These concepts are described by using terms discovered in the text.

Usually text mining is used to indicate a text search technique. However we think of text mining as having more functions. Text mining technologies extract information such as facts, author's intentions, their expectations and their claims which are more useful than keywords in texts. This knowledge is helpful to many applications such as marketing, trend analysis and claim processing.

Research Items

Concept Extraction from Texts

Information extraction from texts is performed by applying natural language processing technology to a s[illegible]c domain. Our system selects keywords and considers synonyms and word sense ambiguities. The system[illegible] a synonym dictionary to equate synonymous words and phrases such as PC and Personal computer. An attribute is associated with each word or phrase to distinguish the meaning. For example we may use Washington (person



Text Mining - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
アドレス(D) http://semcode.trl.ibm.com/IBMfair/index_e.html
移動
IBM Tokyo Research Laboratory
DATA
ページが表示されました
インターネット

Text Mining

Overview

Text mining is used to indicate a text search technique. Text mining technologies extract information such as facts, author's intentions, their expectations and their claims which are more useful than keywords in texts.

Research Items

Concept Extraction from Texts

Our system selects keywords and considers synonyms and word sense ambiguities. The system uses a dictionary to equate synonymous words and phrases such as PC and Personal computer.

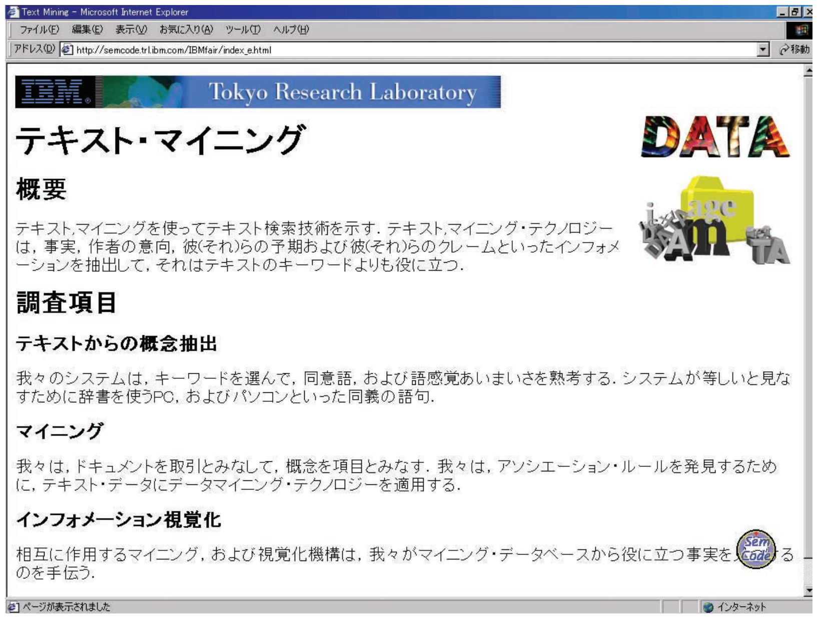
Mining

We regard a document as a transaction and regard a concept as an item. We apply data mining technology to the text data to discover association rules.

Information Visualization

The interactive mining and visualization mechanism helps us to find useful facts from the mining databas

Figure 9. Original (top) and summarized (bottom) documents.


Text Mining - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
アドレス(D) http://semcode.trl.ibm.com/IBMfair/index_e.html
移動
IBM Tokyo Research Laboratory
DATA
ページが表示されました
インターネット

テキスト・マイニング

概要

テキスト,マイニングを使ってテキスト検索技術を示す. テキスト,マイニング・テクノロジーは, 事実, 作者の意向, 彼(それ)らの予期および彼(それ)らのクレームといったインフォメーションを抽出して, それはテキストのキーワードよりも役に立つ.

調査項目

テキストからの概念抽出

我々のシステムは, キーワードを選んで, 同意語, および語感覚あいまいさを熟考する. システムが等しいと見なすために辞書を使うPC, およびパソコンといった同義の語句.

マイニング

我々は, ドキュメントを取引とみなして, 概念を項目とみなす. 我々は, アソシエーション・ルールを発見するために, テキスト・データにデータマイニング・テクノロジーを適用する.

インフォメーション視覚化

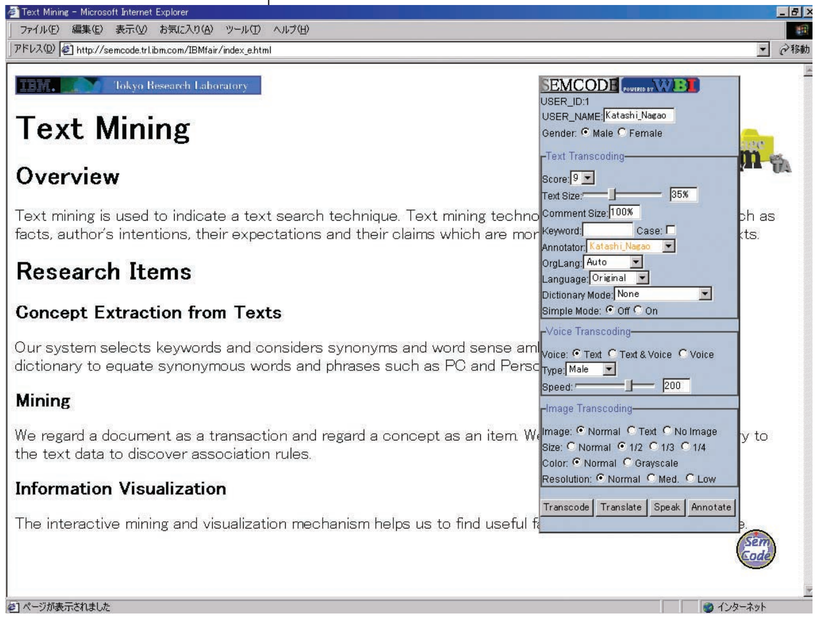
相互に作用するマイニング, および視覚化機構は, 我々がマイニング・データベースから役に立つ事実を見つけるのを手伝う.

Figure 10. Screen shot of a translated document.

inal and the bottom one is the summarized version.

Another kind of text transcoding is language translation. We can predict that translation based on linguistic annotations will produce a much better result than many existing systems. This is because the major difficulties of present machine translation come from syntactic and word sense ambiguities in natural languages, which we can easily clarify in annotation. We show an example of the result of English-to-Japanese translation in Figure 10.

Figure 11. Screen shot of a document that has gone through image transcoding. The preference-setting subwindow appears on the right.


Text Mining - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
アドレス(D) http://semcode.trl.ibm.com/IBMfair/index_e.html
移動
IBM Tokyo Research Laboratory
SEMCODE POWERED BY WBI
Transcode
Translate
Speak
Annotate
Sem Code
ページが表示されました
インターネット

Text Mining

Overview

Text mining is used to indicate a text search technique. Text mining techno ch as facts, author's intentions, their expectations and their claims which are mor xts.

Research Items

Concept Extraction from Texts

Our system selects keywords and considers synonyms and word sense aml dictionary to equate synonymous words and phrases such as PC and Perso

Mining

We regard a document as a transaction and regard a concept as an item. We y to the text data to discover association rules.

Information Visualization

The interactive mining and visualization mechanism helps us to find useful f e.

USER_ID:1
USER_NAME: Katashi_Nagao
Gender: [✓ Male] Female

Text Transcoding
Score: 9
Text Size: 35%
Comment Size: 100%
Keyword:
Case: ☐
Annotator: Katashi_Nagao
OrgLang: Auto
Language: Original
Dictionary Mode: None
Simple Mode: [✓ Off] On

Voice Transcoding
Voice: [✓ Text] Text & Voice Voice
Type: Male
Speed: 200

Image Transcoding
Image: [✓ Normal] Text No Image
Size: Normal [✓ 1/2] 1/3 1/4
Color: [✓ Normal] Grayscale
Resolution: [✓ Normal] Med. Low

Image transcoding

Image transcoding converts images into different sizes, colors (full color or gray scale), and resolutions (for example, the compression ratio) depending on a user's device and communication capability. Links to these converted images are made from the original images. Therefore, users will notice that the images they're viewing aren't original if there are links to similar images.

Figure 11 shows a document that's summarized at half the original size with images reduced to one-third the original size. We show the preference setting subwindow on the right-hand side of the figure. The window appears when the user double-clicks the icon on the lower right corner (the transcoding proxy automatically inserts the icon). Using this window, the user can easily modify the parameters for transcoding. For example, by combining image and text transcodings, the system can convert contents to fit the client's screen size.

Voice transcoding

Voice synthesis also works better if the content has linguistic annotation. For example, some Web engineers are discussing a speech synthesis markup language (see http://www.cstr.ed.ac.uk/projects/ssml.html). A typical example is processing proper nouns and technical terms. Word-level annotations on proper nouns let transcoders recognize not only their meanings but also their readings.

Voice transcoding generates the spoken-language version of documents. Two types of voice transcoding exist. One is when the transcoder synthesizes sound data in audio formats, such as MPEG-1 audio layer 3 (MP3). This case is useful for devices without voice synthesis capability, such as cellular phones and personal digital assistants (PDAs). The other type is when the transcoder converts documents into a more appropriate style for voice synthesis. This case requires that we install a voice-synthesis program on the client side. Of course, the synthesizer uses the output of the voice transcoder. Therefore, the mechanism of document conversion is a common part to both types of voice transcoding.

Documents annotated for voice include some text in commentary annotation for nontextual elements and some word information in linguistic annotation for the reading of proper nouns and unknown words in the dictionary. A document may also contain phrase and sentence boundary information so that pauses appear in the appropriate positions.

Figure 12 shows an example of the voice-transcoded document where we inserted icons that represent the speaker. When the user clicks on the speaker icon, it invokes the MP3 player software and starts playing the synthesized voice data.

Video transcoding

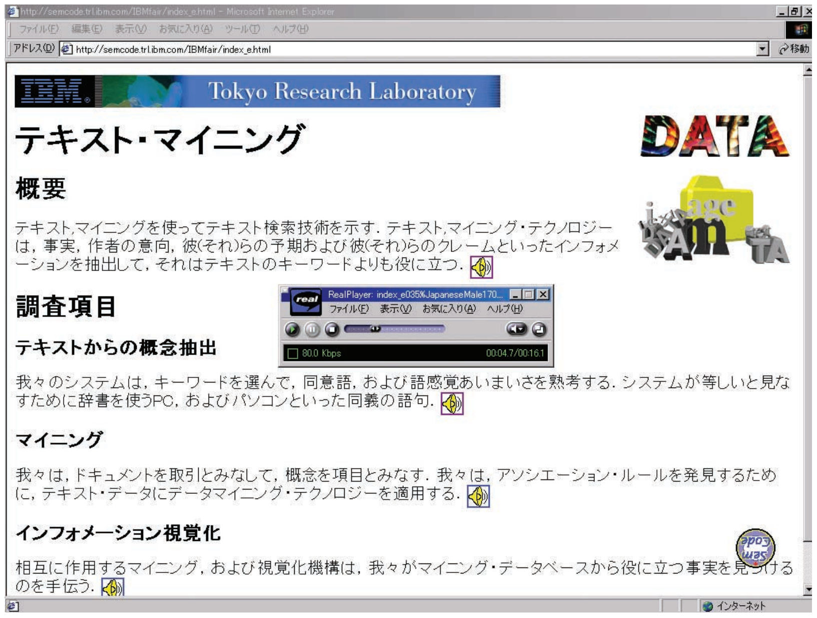
Video transcoding employs video annotation that consists of linguistically marked up transcripts—such as closed captions, time stamps of scene changes, representative images (key frames) of each scene—and additional information such as program names. Our video transcoding has several variations, including video summarization, video-to-document transformation, video translation, and so on.

We perform video summarization as a by-product of text summarization. Since a summarized video transcript contains important information, corresponding video sequences will produce a collection of significant scenes in the video. We developed a video player that plays summarized videos. Figure 13 shows a screen shot of our video player.

To play the summarized video, the player accepts a Synchronized Multimedia Integration Language (SMIL) file (see http://www.w3.org/TR/REC-smil/), which specifies the beginning and ending times of each clip in `<anchor>` tags. We extended the `<anchor>` tag specification slightly with an additional attribute, `insummary`, which tells whether a particular clip is in the previously created summary. Here's a sample of the SMIL format with our extension.

```
<smil>
<head> ... </head>
<body>
<video region="..."
     src="somefile.mpg">
  <anchor title="..."
   begin="00:00:02"
   end="00:05:24"/>
  <anchor title="Web... "
   begin="00:20:08"
   end="00:38:01" insummary="true"/>
  ...
</body>
</smil>
```

There's been previous work performed on video summarization—for example, Infomedia[10] and CueVideo.[11] These programs create a video summary based on automatically extracted features in video such as scene changes; speech, text, and human faces in frames; and closed captions. Video summarization applications also transcode the video data without annotations. Currently, however, a quality of their summarization isn't practical because of the failure of automatic video analysis. Automatic scene detection sometimes extracts unnecessary scenes. Also, automatic speech recognition often misrecognizes words. These results then affect noisy summaries. Our approach to video summarization is sufficient if the data has enough semantic annotation. Our tool helps annotators create semantic annotation data for multimedia data. Since our annotation data is task independent and versatile, annotations on video are worth creating if the video will be used in different applications (such as automatic editing and information extraction from video).


http://semcode.trl.ibm.com/IBMfair/index_e.html - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
アドレス(D) http://semcode.trl.ibm.com/IBMfair/index_e.html
移動
IBM Tokyo Research Laboratory
DATA
RealPlayer: index_e035%JapaneseMale170...
ファイル(F) 表示(V) お気に入り(A) ヘルプ(H)
80.0 Kbps 00:04.7/00:16.1
インターネット

テキスト・マイニング

概要

テキスト,マイニングを使ってテキスト検索技術を示す. テキスト,マイニング・テクノロジーは, 事実, 作者の意向, 彼(それ)らの予期および彼(それ)らのクレームといったインフォメーションを抽出して, それはテキストのキーワードよりも役に立つ.

調査項目

テキストからの概念抽出

我々のシステムは, キーワードを選んで, 同意語, および語感覚あいまいさを熟考する. システムが等しいと見なすために辞書を使うPC, およびパソコンといった同義の語句.

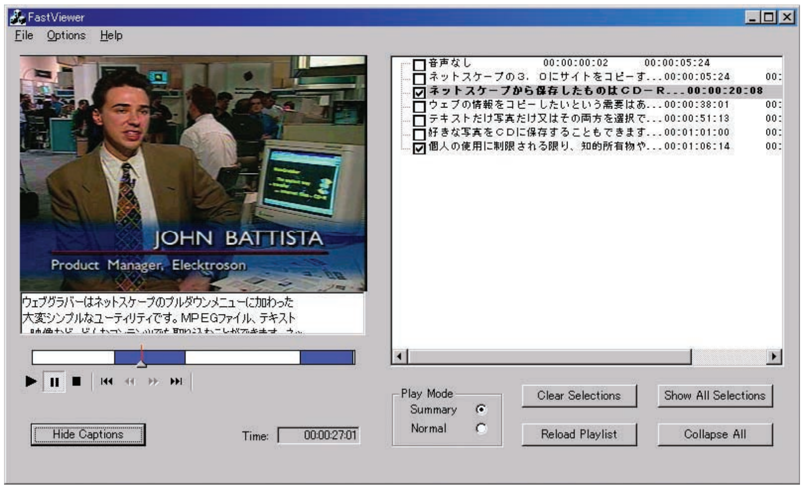
マイニング

我々は, ドキュメントを取引とみなして, 概念を項目とみなす. 我々は, アソシエーション・ルールを発見するために, テキスト・データにデータマイニング・テクノロジーを適用する.

インフォメーション視覚化

相互に作用するマイニング, および視覚化機構は, 我々がマイニング・データベースから役に立つ事実を見つけるのを手伝う.

Figure 12. Screen shot of a voice transcoded document.



ウェブグラバーはネットスケープのプルダウンメニューに加わった
大変シンプルなユーティリティです。MPEGファイル、テキスト

☐	音声なし	00:00:00:02	00:00:05:24
☐	ネットスケープの3．0にサイトをコピーす...	00:00:05:24	00:
✓	**ネットスケープから保存したものはCD－R...**	**00:00:20:08**	
☐	ウェブの情報をコピーしたいという需要はあ...	00:00:38:01	00:
☐	テキストだけ写真だけ又はその両方を選択で...	00:00:51:13	00:
☐	好きな写真をCDに保存することもできます...	00:01:01:00	00:
✓	個人の使用に制限される限り、知的所有物や...	00:01:06:14	00:

Play Mode: [✓ Summary] Normal

Time: 00:00:27:01

Figure 13. Screen shot of video player with the summarization function.

Video-to-document transformation is another type of video transcoding. If the client device doesn't have video playing capability, the user can't access video contents. In this case, the video transcoder creates a document including important images of scenes and texts related to each scene. The text transcoder can also summarize the resulting document.

Our system implements two types of video translation. One is a translation of automatically generated subtitle text. The transcript with time codes generates the subtitle text. The format of the text looks like

```
<subtitle duration="00:01:19">
<time begin="00:00:00"/>No speech
<clear/>
<time begin="00:00:05"/>....
<time begin="00:00:07"/>....
<time begin="00:00:12"/>....
<clear/>
....
</subtitle>
```

The text transcoder can translate the subtitle text into different languages as the user wants, and the video player shows the results synchronized with the video.

The other type of video translation uses a combination of text and voice transcodings. First, the text transcoder translates a video transcript with linguistic annotation. Then, the voice transcoder converts the translation's result into voice-suitable text. Synchronization of video playing and voice synthesis makes another language version of the original video clip. The video player adjusts the duration of each voice data according to the length of its corresponding video segment by changing the speed of the synthesized voice.

Using object-level annotations, the video transcoder can create an interactive hyperlinked-video in which video objects link to information such as names, times, locations,[12] related Web sites, and so on. The transcoder uses the annotations for retrieving objects from multiple video clips and generating object-featured video summaries. For example, a user might watch a news clip showing Alan Greenspan's (Chairman of the US Federal Reserve Board) comments on the economy. To find out who Greenspan is, the user might pause the video and click on his face to find his biography or a list of links to other text or video clips concerning his announcements and their effect on the economy.

Our system automatically combines the text, image, voice, and video transcodings are automatically combined according to user demand. This gives the transcoding proxy a planning mechanism to determine the order of activation of each transcoder necessary for the requested content and user preferences, including client device constraints.

Future plans

We plan to apply our technology to knowledge discovery from huge online resources. Annotations will help extract some essential points in documents. For example, if an annotator adds comments to several documents, and seems to be a specialist of a particular field, then the machine automatically collects documents annotated by this annotator and generates a single document—including summaries of the annotated documents.

We're also pursuing content-based retrieval of Web documents, including multimedia data. Such retrieval lets users ask questions in natural language (either spoken or written). We imagine that in the near future we'll not use search engines; instead, we'll use knowledge discovery engines that give us a personalized summary of multiple documents—not hyperlinks.

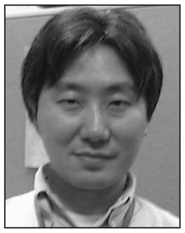
While our current prototype system is running locally, we're also planning to evaluate our system with open experiments with some universities in Japan. Once we've evaluated our system, we'll distribute our annotation editor—with natural language processing capabilities—for free. This is one step toward dealing with the oncoming information deluge. **MM**

Acknowledgments

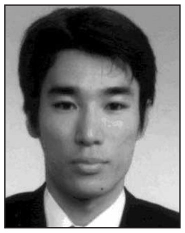
We'd like to acknowledge the contributors to this project. Shingo Hosoya and Ryuichiro Higashinaka developed the prototype of some plug-ins of the semantic transcoder. Koiti Hasida gave us some helpful advice to apply the GDA tag set to our linguistic and multimedia annotations. Paul Maglio and his group at IBM Almaden Research Center provided us with the WBI developer's kit. Hideo Watanabe developed the prototype of the English-to-Japanese language translator. Masashi Nishimura provided us with the speech recognition software developer's kit. Shinichi Torihara developed the voice synthesizer prototype.

References

1. M. Hori et al., “Annotation-based Web Content Transcoding,” *Proc. Ninth Int’l WWW Conf.*, Elsevier, Amsterdam, 2000, pp. 197-211.
2. P. Brusilovsky, E. Schwarz, and G. Weber, “A Tool for Developing Adaptive Electronic Textbooks on WWW,” *Proc. World Conf. of the Web Soc.* (WebNet 96), 1996, pp. 64-69.
3. M. Milosavljevic et al. “Virtual Museums on the Information Superhighway: Prospects and Potholes,” *Proc. Ann. Conf. Int’l Committee for Documentation of the International Council of Museums* (CIDOC 98), 1998, pp. 10-14.
4. J. Fink, A. Kobsa, and A. Nill, “Adaptable and Adaptive Information Provision for All Users, Including Disabled and Elderly People,” *New Review of Hypermedia and Multimedia 4*, 1998, pp. 163-188, http://www.zeus.gmd.de/ kobsa/papers/1998-NRMH-kobsa.ps.
5. H. Maruyama, K. Tamura, and N. Uramoto, *XML and Java: Developing Web Applications*, Addison-Wesley, Reading, Mass., 1999.
6. H. Watanabe, *Linguistic Annotation Language: The Markup Langauge for Assisting NLP Programs*, TRL research report RT0334, IBM Tokyo Research Lab, 1999.
7. M. Roscheisen, C. Mogensen, and T. Winograd, *Shared Web Annotations as a Platform for Third-Party Value-Added Information Providers: Architecture, Protocols, and Usage Examples,* tech. report CSDTR/DLTR, Computer Science Dept., Stanford Univ., Stanford, Calif., 1995.
8. M.A. Schickler, M.S. Mazer, and C. Brooks, “Pan-Browser Support for Annotations and Other Meta-Information on the World Wide Web,” *Computer Networks and ISDN Systems*, vol. 28, 1996.
9. K. Nagao and K. Hasida, “Automatic Text Summarization Based on the Global Document Annotation,” *Proc.Int’l Conf. Computational Linguistics* (COLING-ACL 98), Morgan Kaufmann, San Francisco, 1998, pp. 917-921.
10. M.A. Smith and T. Kanade, *Video Skimming for Quick Browsing Based on Audio and Image Characterization*, tech. report CMU-CS-95-186, School of Computer Science, Carnegie Mellon Univ., Pittsburgh, Pa., 1995.
11. A. Amir et al., “CueVideo: Automated Indexing of Video for Searching and Browsing,” *Proc. 22nd Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval* (SIGIR 99), ACM Press, New York, 1999.
12. M.G. Christel, A.M. Olligschlaeger, and C. Huang, “Interactive Maps for a Digital Video Library,” *IEEE MultiMedia*, vol. 7, no. 1, Jan.-Mar. 2000, pp. 60-67.



Katashi Nagao received his BE, ME, and PhD degrees in computer science from the Tokyo Institute of Technology in 1985, 1987, and 1994, respectively. Since 1987, he has been researching natural language processing and machine translation systems at IBM Research, Tokyo Research Laboratory. He has conducted research projects on natural language dialogue, multiagent systems, and human–computer interaction at Sony Computer Science Labs. He rejoined the IBM Tokyo Research Laboratory in 1999, where he is currently working on semantic transcoding and semantic discovery projects.



Yoshinari Shirai received his BE and ME degrees from the Graduate School of Media and Governance, Keio University in 1998 and 2000, respectively. He’s currently a researcher at the NTT Communication Science Laboratories. His research interests include human-computer interaction and mobile computing.

Kevin Squire received his BS in computer engineering from Case Western Reserve University in Cleveland, Ohio, and his MS in electrical engineering from the University of Illinois at Urbana-Champaign (UIUC). He’s currently a PhD candidate in electrical engineering at UIUC, studying multimodal language acquisition. His research interests include visual pattern recognition and speech and language processing

Readers may contact Nagao at IBM Research, Tokyo Research Laboratory, Kanagawa, Japan, email knagao@jp.ibm.com.