Semantic Annotation and Intelligent Content

Proceedings of the Workshop
Supported by SIGLEX, the ACL Special Interest Group
on the Lexicon

Paul Buitelaar, Kôiti Hasida
Editors

August 5th/6th, 2000
Centre Universitaire
Luxembourg

COLING 2000

Topics and Motivation

SEMANTIC ANNOTATION is augmentation of data to facilitate automatic recognition of the underlying semantic structure. A common practice in this respect is labeling of documents with thesaurus classes for the sake of document classification and management. In the medical domain, for instance, there is a long-standing tradition in terminology maintenance and annotation/classification of documents using standard coding systems such as ICD, MeSH and the UMLS meta-thesaurus. Semantic annotation in a broader sense also addresses document structure (title, section, paragraph, etc.), linguistic structure (dependency, coordination, thematic role, co-reference, etc.), and so forth. In NLP, semantic annotation has been used in connection with machine-learning software trainable on annotated corpora for parsing, word-sense disambiguation, co-reference resolution, summarization, information extraction, and other tasks. A still unexplored but important potential of semantic annotation is that it can provide a common I/O format through which to integrate various component technologies in NLP and AI such as speech recognition, parsing, generation, inference, and so on.

INTELLIGENT CONTENT is semantically structured data that is used for a wide range of content-oriented applications such as classification, retrieval, extraction, translation, presentation, and question-answering, as the organization of such data provides machines with accurate semantic input to those technologies. Semantically annotated resources as described above are typical examples of intelligent content, whereas another major class includes electronic dictionaries and inter-lingual or knowledge-representation data. Some ongoing projects along these lines are GDA (Global Document Annotation), UNL (Universal Networking Language) and SHOE (Simple HTML Ontology Extension), all of which aim at motivating people to semantically organize electronic documents in machine-understandable formats, and at developing and spreading content-oriented application technologies aware of such formats. Along similar lines, MPEG-7 is a framework for semantically annotating audiovisual data for the sake of content-based retrieval and browsing, among others. Incorporation of linguistic annotation into MPEG-7 is in the agenda, because linguistic descriptions already constitute a main part of existing metadata. In short, semantic annotation is a central, basic technology for intelligent content, which in turn is a key notion in systematically coordinating various applications of semantic annotation. In the hope of fueling some of the developments mentioned above and thus promoting the linkage between basic researches and practical applications, the workshop invites researchers and practitioners from such fields as computational linguistics, document processing, terminology, information science, and multimedia content, among others, to discuss various aspects of semantic annotation and intelligent content in an interdisciplinary way.

Paul Buitelaar, Kôiti Hasida

Organisation and Invited Talks

Organisers

Paul Buitelaar
Kôiti Hasida

Program Committee

Amit Bagga, GE Corporate R&D, USA
Paul Buitelaar, DFKI-LT, Germany (Co-Chair)
Gregor Erbach, FTW, Austria
Christiane Fellbaum, Princeton University, USA
Wolfgang Giere, ZINFO, University of Frankfurt, Germany
Nicola Guarino, Ladseb-CNR Padova, Italy
Kôiti Hasida, ETL, Japan (Co-Chair)
Boris Katz, AI Laboratory, MIT, USA
Adam Kilgarriff, University of Brighton, UK
Elizabeth Liddy, Syracuse University, USA
Katashi Nagao, IBM TRL, Japan
Hiroshi Nakagawa, University of Tokyo, Japan
Hwee Tou Ng, DSO, Singapore
Martha Palmer, University of Pennsylvania, USA
Virach Sornlertlamvanich, NECTEC, Thailand
Steffen Staab, University of Karlsruhe, Germany
Henry Thompson, Edinburgh University, UK
Hiroshi Uchida, United Nations University, Japan
Rémi Zajac, CRL, New Mexico State University, USA

Invited Talks

In-depth Utilization of Natural Language Processing for Rich Semantic Annotation
Elizabeth Liddy, Syracuse University, USA
MindNet as a Framework for Semantically Structuring Text
Bill Dolan, Microsoft, USA
UNL: Interlingua as Intelligent Content
Hiroshi Uchida, United Nations University, Japan
GDA: Semantically Annotated Documents as Intelligent Content
Koiti Hasida, ETL, Japan

Table of Contents

Workshop Papers

SECTION 1

Semantic Annotation of Word Class and Dependency Structure

Semantic annotation of a Japanese speech corpus

John Fry
Linguistics Dept. & CSLI
Stanford University
Stanford CA 94305-2150 USA
fry@csli.stanford.edu

Francis Bond[*]
Machine Translation Research Group
NTT Communication Science Laboratories
2-4 Hikari-dai, Kyoto 619-0237 JAPAN
bond@cslab.kecl.ntt.co.jp

Abstract

This paper describes the semantic annotations we are performing on the **CallHome Japanese** corpus of spontaneous, unscripted telephone conversations (LDC, 1996). Our annotations include (i) semantic classes for all nouns and verbs; (ii) verb senses for all main verbs; and (iii) relations between main verbs and their complements in the same utterance. Our semantic tagset is taken from NTT's **Goi-Taikei** semantic lexicon and ontology (Ikehara et al., 1997). A pilot study demonstrates that the verb sense tagging can be efficiently performed by native Japanese speakers using computer-generated HTML forms, and that good inter-annotator reliability can be obtained in the right conditions.

1 Introduction

Semantic annotations have proved valuable for a variety of NLP tasks, including parsing, word sense disambiguation, coreference resolution, summarization, and information retrieval and extraction. The most challenging domain for all these tasks is *spontaneous spoken language*, which tends to be more terse, less grammatical, less structured, and more ambiguous than planned or written text. For this reason, the annotation of spoken language corpora with accurate, high-quality linguistic tags has become a topic of great interest recently (Dybkjær et al., 1998; Ide, 1998; Core et al., 1999).

The target of our semantic annotations is the CallHome Japanese (CHJ) corpus (LDC, 1996). The CHJ corpus consists of digitized speech data and text transcriptions of 120 spontaneous, unscripted telephone conversations in Japanese. Each transcript is encoded in EUC-format Japanese characters and covers a contiguous 5 or 10 minute segment taken from a recorded conversation lasting up to 30 minutes. To illustrate, a brief fragment (the first three utterances) of a CHJ transcript is given in Figure 1 (an English gloss appears below the fragment). Each utterance in a transcript is analyzed into individual morphemes, with transcriber comments in brackets. The speaker (A or B) and the start and end times of each utterance (i.e. speaker turn) are also provided in the transcripts. The 120 conversations in the CHJ corpus contain a total of about 340,000 word/morpheme tokens, 12,000 unique word/morpheme types, and 39,000 speaker turns.

The CHJ corpus was originally created for research in large vocabulary speech recognition. However, we hope to make the corpus useful for other types of NLP research (by ourselves and others) by supplementing it with a variety of linguistic annotations. When finished, we plan to make the annotated CHJ corpus available to the research community through the LDC.

We are annotating the CHJ corpus with a variety of syntactic, semantic, and acoustic/prosodic tags. In this paper we focus on our semantic annotations, which include the following:

- **Semantic classes** for all verbs and nouns.
- **Verb senses** for each main verb.
- **Predicate-argument relations** between main verbs and their explicitly mentioned verb complements, labeled with thematic roles.

By way of example, Table 1 shows the end time boundary t_e (i.e. duration), POS, pronunciation, canonical form (including verb sense

[*] Visiting CSLI, Stanford University (1999-2000).

```
120.20 123.35 A: 嘘、 @私[[あたし,col]] 職 が なくて さ 日本 に 帰った ら。
123.26 123.70 B: うん。
124.28 128.50 A: トニー の お 誕生日 九月 二日 だ から @何か[[なんか,col]] 買って あげ
よう と 思った けど、 @私[[あたし,col]] さ 無職 の 人間 だ {laugh} 。
```

Translation:

A:	嘘、	私	職	が	なくて	さ	日本	に	帰った	ら。
	Uso,	atashi	shoku	-ga	nakute	sa	nihon	-ni	kaetta	-ra.
	Lie,	I	employment	NOM	not-have	y’know	Japan	DAT	returned	if.

‘No way. Having no employment, y’know, maybe I should return to Japan?’

B:	うん。
	Un.

‘Uh-huh.’

A:	トニー	の	お	誕生日	九月	二日	だ	から	何か	買って	あげ	よう	と
	Tonii	-no	o-	tanjoubi	kugatsu	futsuka	da	kara	nanka	katte	age-	you	-to
	Tony	GEN	HON	birthday	Sept.	2nd	is	because	something	buy	give	VOL	COMP

思った	けど、	私	さ	無職	の	人間	だ。
omotta	kedo,	atashi	sa	mushoku	-no	ningen	da.
think	but,	I	y’know	jobless	GEN	person	am.

‘Tony’s birthday is Sept. 2 so I want to buy him something, but I’m unemployed.’

Figure 1: Fragment from CHJ transcript 0696

ID	Word	t_e (s)	POS	Phonetic	Canonical	Class	Arguments
003	あたし	0.810	pro	atasi	私	c0008	
004	職	1.060	noun	syoku	職	c1939	
005	が	1.180	part	ga			
006	なくて	1.580	v-neg,te	nakute	無い(3)	v0003	N1:003,N2:004
007	さ	1.840	part	sa			
008	日本	2.134	prop	nihoN	日本	c0385,c0463,p0030	
009	に	2.250	part	ni			
010	帰った	2.610	v-r5	kaeqta	帰る(2)	v0014	N1:003,N5:008
011	ら	2.940	cond2	ra			

Table 1: Sample annotations from the first utterance of 0696

number), semantic class, and argument indexes for most of the first utterance from Figure 1.

Our current plan is to provide our annotations in the simple tabular text format shown in Table 1, rather than in one of the of the numerous annotation frameworks currently in contention.[1] If a reliable and widely-accepted XML encoding framework emerges before we release our annotations, then we will consider adopting that scheme. However, our primary aim is to provide simple, accurate, low-level annotations (upon which other, higher-level annotations might be based) so that language researchers can use the corpus more flexibly and with greater confidence.

Some of the annotations which we have already completed, or nearly completed, but will *not* discuss in this paper include the following:

- **Phonetic transcriptions**, in Roman characters (*kunreisiki* transliteration), of all 120 conversations.
- **POS tags** using the LDC’s existing inventory of 60 syntactic and morphological tags for Japanese.

[1] For example, 53 annotation frameworks are listed at `http://morph.ldc.upenn.edu/annotation/`. For speech corpora, notable contributions include the Corpus Encoding Standard (CES, `http://www.cs.vassar.edu/CES/`), MATE (Dybkjær et al., 1998), and the ‘annotation graph’ approach of Bird and Liberman (1998).

- Raw **acoustic data** from the ESPS/waves+ speech processing software, including f_0 (fundamental frequency) and power (root-mean-square amplitude) measurements at 10ms intervals.
- The **duration** of each word, based on semi-automatic word segmentation of the speech data.

The remainder of this paper is organized as follows. In Section 2 we describe NTT's Goi-Taikei semantic dictionary, which is the source of our semantic tagset. Section 3 describes our tagging methodology and our pilot study of the verb sense annotation task. Finally, Section 4 describes our browser-based annotation application.

2 Goi-Taikei

As a base for our tags, we are using the Goi-Taikei (**GT**) Japanese lexicon (Ikehara et al., 1997), a 400,000-word lexicon and ontology developed by NTT for machine translation (MT) applications.

We decided that **GT** is an appropriate resource for our semantic annotation task for three reasons. First, semantic information from **GT** has already proved valuable in a variety of NLP applications in Japan, including parsing, morphological analysis, text-to-speech, proofreading, and MT (Ikehara et al., 1994; Shirai et al., 1995; Akiba et al., 1995; Oku, 1996; Nakaiwa and Seki, 1999; Baldwin et al., 1999; Baldwin and Tanaka, 1999; Yokoyama and Ochiai, 1999). Secondly, the **GT** lexicon and ontology, at 400,000 words, is significantly larger than earlier dictionaries, such as the 260,000-word EDR Dictionary (EDR, 1996) and the 2,000-word IPAL lexicon (IPA, 1987; IPA, 1990; IPA, 1996). **GT** also contains detailed valency information for 16,000 predicate senses, which makes it more suited to our task than the Kadokawa thesaurus (Hamanishi and Ono, 1990). Finally, **GT** is available in book and CD-ROM format at a price (around US $750) that is several times lower than EDR.

GT consists of three main components: (i) an ontology, (ii) a semantic word dictionary, and (iii) a semantic structure dictionary which includes subcategorization frames for verbs and adjectives.

2.1 Ontology

GT's ontology classifies concepts to use in expressing relationships between words. The meanings of common nouns are given in terms of a semantic hierarchy of 2,710 nodes. Most of the top four levels of the semantic hierarchy are shown in Figure 2, with two examples of deeper nodes. Each node represents a semantic class. Edges in the hierarchy represent **is-a** or **has-a** relationships, so that the child of a semantic class related by an **is-a** relation is subsumed by it. For example, `nation` **is-a** `organization`. In addition to the 2,710 classes (12-level tree structure) for common nouns, there are 200 classes (9-level tree structure) for proper nouns and 108 classes (5-level tree structure) for predicates.

2.2 Semantic Word Dictionary

The **GT** semantic dictionary includes 100,000 common nouns, 200,000 proper nouns, 70,000 technical terms and 30,000 other words: 400,000 words in all.

Figure 3 shows a simplified example of one record of the Japanese semantic word dictionary. Each record specifies an index form, pronunciation, canonical form, syntactic information and a set of semantic classes. The syntactic information includes the part of speech, inflectional class, detailed parts of speech, conjunctive conditions and so on. Each word can have up to five common noun semantic classes and ten proper noun semantic classes. The numbering system gives common-noun semantic classes a prefix of `c`, proper-noun classes a prefix of `p`, and predicate classes a prefix of `v`. In Figure 3, for example, the word 日本 *nihon* "Japan" belongs to the common-noun classes `c0385 nation` (⊂ `organization`) and `c0463 territory` (⊂ `place`), and to proper-noun class `p0030 country` (⊂ `place name`). More examples of semantic classes for the nouns and verbs from the annotated CHJ fragment in Table 1 are listed in that table under the column labeled 'Class'.

2.3 Semantic Structure Dictionary

The basic structure of a clause comes from the relationship between the main verb and nouns. **GT**'s structure transfer dictionary, designed for MT applications, provides this basic clause structure. **GT** provides 10,000 patterns in its common structure transfer dictionary and

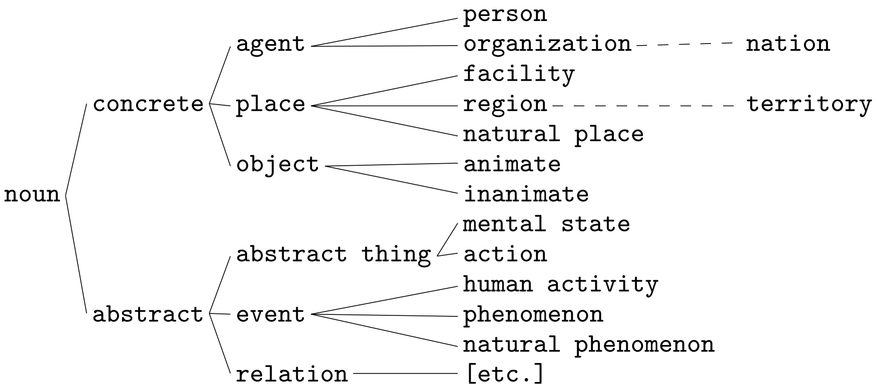


Figure 2: **GT**'s Semantic Hierarchy (top 4 levels)

Index Form	日本		
Pronunciation	/nihon/		
Canonical Form	日本		
Part of Speech	`noun`		
Semantic Classes	common noun	`nation` (c0385), `territory` (c0463)	
	proper noun	`country` (p0030)	

Figure 3: Japanese Lexical Entry for noun 日本 *nihon* "Japan"

5,000 patterns in its idiomatic structure transfer dictionary. The common structure transfer dictionary contains an average of 2.3 patterns for each verb.

Figure 4 gives an example from the common structure transfer dictionary. Each predicate is associated with one or more arguments labeled N1, N2, ... Each case-slot contains information such as grammatical function, case-marker, case-role, semantic restrictions on the filler and default order (not all features are shown in the example). The arguments correspond between Japanese and English, thus giving the backbone of the transfer.

2.3.1 Case Roles

Case-elements in the valency dictionary are associated with particular case roles (also known as thematic roles, θ-roles, or deep cases). The current set of case roles is given in Table 2, along with the most commonly associated case markers[2] in Japanese, and prepositions or grammatical functions (gf) in English. The annotated CHJ fragment in Table 1 shows some specific case-role fillers under the column labeled 'Arguments'. For example, word 008, 日本 *nihon* "Japan", serves as the goal argument (`N5`) of the verb 帰った *kaetta* "returned (home)" in that utterance.

There seems to be no consensus among linguists on what the best set of case roles is, or even whether case roles should be replaced by more abstract primitives or more concrete participant-roles. In any case, case roles have in practice proved extremely useful for NLP and are used in most MT systems (Bond and Shirai, 1997).

3 Tagging methodology

We are annotating the CHJ corpus with (i) semantic classes for all nouns and verbs; (ii) verb senses for all main verbs; and (iii) predicate-argument relations between main verbs and their complements in the same utterance. Our tagging of verb senses and predicate-argument relations relies on the browser-based annotation application described in Section 4. The predicate-argument tags, based on **GT**'s semantic structure dictionary (Section 2.3), provide a basic dependency structure for each utterance.

[2] Japanese case markers (also known as 'particles') are postpositions, and are similar to English prepositions in many ways.

$$\begin{bmatrix}
\text{Pattern-ID} & \text{-0002-00-} \\
\text{Semantic Class} & (\texttt{action}) \\
\text{Japanese} & \begin{bmatrix} \text{pred} & \text{取る}\,\textit{toru}\text{ (verb)} \\ \text{N1} & \begin{bmatrix} \text{case-marker} & \text{が }\textit{ga}\text{ “}\textsc{nom}\text{” (Agent)} \\ \text{restriction} & \texttt{agent} \end{bmatrix} \\ \text{N2} & \begin{bmatrix} \text{case-marker} & \text{を }\textit{o}\text{ “}\textsc{acc}\text{” (Object-1)} \\ \text{restriction} & \texttt{lodging, room, vehicle, ...} \end{bmatrix} \end{bmatrix} \\
\text{English} & \begin{bmatrix} \text{pred} & \textit{reserve}\text{ (verb)} \\ \text{N1} & \begin{bmatrix} \text{function} & \text{subject (nominative)} \end{bmatrix} \\ \text{N2} & \begin{bmatrix} \text{function} & \text{direct-object (accusative)} \end{bmatrix} \end{bmatrix}
\end{bmatrix}$$

Figure 4: Part of the common structure transfer dictionary for one sense of 取る *toru* “take”

Label	Name	Case-marker	English	Label	Name	Case-marker	English
N1	Agent	ga (kara, towa)	Subj (gf)	N8	Locative	ni, o, de, e, kara	in/at/on
N2	Object-1	o (nituite)	Obj (gf)	N9	Comitative	to	with
N3	Object-2	ni (...)	I-Obj (gf)	N10	Quotative	to	
N4	Source	kara, yori	from	N11	Material	kara, yori, de	with, from
N5	Goal	ni, e, made	to (until)	N12	Cause	kara, yori, de	for
N6	Purpose	ni	for	N13	Instrument	de	with
N7	Result	ni, to	as	N14	Means	de	by

Table 2: Case-roles

Although spontaneous utterances like those in the CHJ corpus are often fragmentary and ungrammatical, rendering full syntactic parsing impractical, the basic relations between predicates and their arguments still hold.

3.1 Tagging semantic classes

We assign **GT** semantic classes to individual CHJ nouns and verbs by automatic table lookup on their **GT** canonical form. In both **GT** and the CHJ lexicon, the canonical forms of words are generally in Chinese characters (*kanji*). For example, the annotated CHJ fragment in Table 1 shows some canonical forms under the column labeled ‘Canonical’. For the verbs in Table 1, the canonical (dictionary) form includes the **GT** verb sense number.

In the majority of cases, the **GT** canonical form of a verb or noun is identical to the canonical form which appears in the CHJ lexicon. The small percentage of cases where the canonical forms differ are corrected by hand. For the approximately 700 nouns in the CHJ corpus that are not covered in the **GT** lexicon (mainly personal names), we assign the closest available **GT** class(es) by hand.

We are marking each noun and verb in the corpus with *all* of its **GT** semantic classes, even those which might be inappropriate for the word in its particular utterance context. For example, in the annotated CHJ fragment in Table 1, the noun 日本 *nihon* “Japan” is marked with all three of its **GT** classes: `c0385 nation` (⊂ `organization`), `c0463 territory` (⊂ `place`), and `p0030 country` (⊂ `place name`). Naturally, it would preferable to exclude those semantic classes which are inappropriate for a given noun or verb in its particular context of use in the CHJ corpus. However, this would require human coders to classify hundreds of thousands of word tokens based on the perceived context in the conversation. In addition, it would be hard to obtain high inter-annotator reliability, given the context-dependent nature of the task and the amount of overlap in the semantic classes.

3.2 Tagging verb senses and arguments

We are providing human-tagged verb sense and verb argument annotations for each main verb in the corpus. Auxiliary verbs and other forms of verb morphology are ignored, except in cases like the passive and causative in which the valence of the main verb is altered. In those cases, special passive or causative senses are provided.

Our plan is to annotate the **GT** verb senses and argument indexes according to the majority judgments of three native-speaker student assistants. The students will make the annotations by clicking on menu choices in a web application that we generate automatically from the **GT** dictionary files and CHJ transcript files (Section 4).

3.3 A pilot study

In preparation for the verb sense tagging project, we conducted a pilot study in which we asked five native speakers to select **GT** verb senses and identify intrasentential arguments for all 110 main verbs in one five-minute CHJ transcript.

Our initial results showed that pairwise inter-annotator agreement on verb senses was 0.68. When chance agreement is taken into account via the kappa statistic, the result, $\kappa = 0.63$, shows that annotator agreement was not reliable (Carletta, 1996). However, we discovered that this result was largely attributable to the annotators' selection of the category "none (of the above)," which the five judges picked with highly variable frequency ($\{7, 11, 22, 25, 26\}, s = 8.6$). For 51 of the 110 verb tokens (46%), "none" was selected by one or more judges, and agreement was low (0.48, $\kappa = 0.42$). When the "none" answers were disregarded, pairwise agreement on those verbs rose considerably (to 0.67, $\kappa = 0.64$). For the remaining 59 verb tokens (54%), "none" was never chosen and pairwise agreement was very reliable (0.84, $\kappa = 0.82$). For the second task, identifying the intrasentential arguments to verbs, pairwise agreement was also very high: 0.89 among annotators who chose the same verb sense.

We then examined more closely those cases in which the low agreement was attributable to inconsistent use of the category "none". We found that most of these cases involved very common, 'light' verbs such as する *suru* "do" and なる *naru* "become". As it turned out, **GT** was lacking some common colloquial senses for these verbs. For example, the verb する *suru* "do" was often used as in utterance (1).

(1)	後	２週間	した	ら
	ato	ni-shuu-kan	shita	ra
	after	2-weeks-long	did	if

'in about two weeks'

This sense of する *suru* "do" does not appear in the **GT** lexicon. In written Japanese *suru* would not normally be used in this way; rather, a more specialized verb such as 経つ *tatsu* "pass" would be preferred.

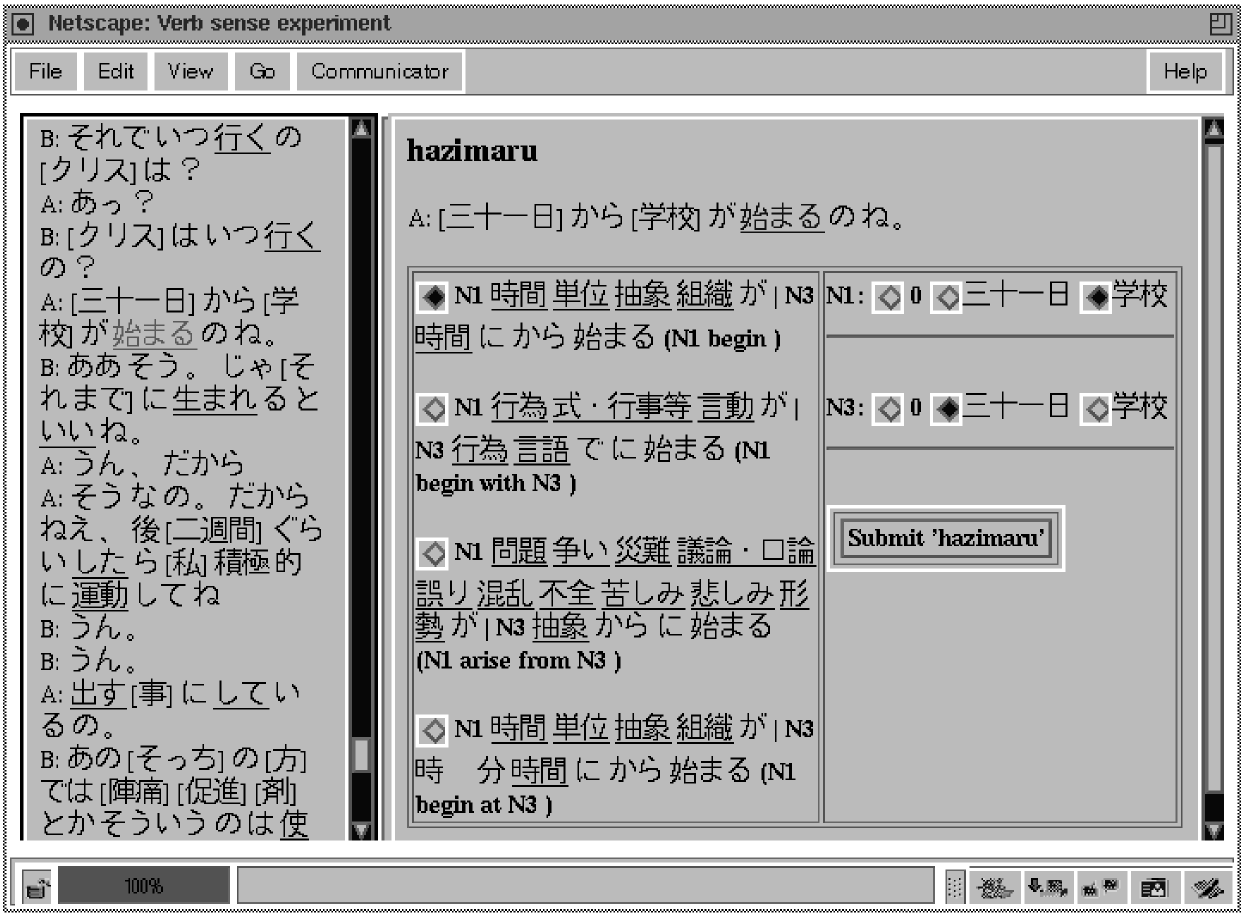
In sum, the results of our pilot study lead us to conclude that we are likely to obtain reliable inter-annotator agreement on the verb sense task, provided the following steps are taken:

- The highly general, spoken-language senses of certain common 'light' verbs need to be added to **GT**.
- The conditions under which the coders are to use the category "none (of the above)" need to be carefully delineated. In particular, "none" should only be selected when there is a particular, standard, well-defined sense of the verb in question that clearly should be listed in the lexicon but is not.

We are encouraged in this regard by the results of Kilgarriff and Rosenzweig (1999) who were able, using careful experimental methodology, to achieve replicable agreement on English word senses (albeit by lexicographers, not students) of 95% in the SENSEVAL project. Our experience echoes the observation of Kilgarriff (1998) that annotators should be given the opportunity to offer feedback to the lexicographers, including information such as inadequate or missing verb senses.

4 The annotation application

We developed a browser-based semantic annotation application specifically for this project. The application is implemented using HTML forms within a web browser (Figure 5). We wrote Perl scripts to generate the annotation forms automatically, using the CHJ transcripts, CHJ lexicon, and **GT** database as input.



B: それでいつ行くの[クリス]は？
A: あっ？
B: [クリス]はいつ行くの？
A: [三十一日]から[学校]が始まるのね。
B: ああそう。じゃ[それまで]に生まれるといいね。
A: うん、だから
A: そうなの。だからねえ、後[二週間]ぐらいしたら[私]積極的に運動してね
B: うん。
B: うん。
A: 出す[事]にしているの。
B: あの[そっち]の[方]では[陣痛][促進][剤]とかそういうのは使

hazimaru

A: [三十一日]から[学校]が始まるのね。

◆ **N1** 時間 単位 抽象 組織 が | **N3** 時間 に から 始まる **(N1 begin)**

◇ **N1** 行為 式・行事等 言動 が | **N3** 行為 言語 で に 始まる **(N1 begin with N3)**

◇ **N1** 問題 争い 災難 議論・口論 誤り 混乱 不全 苦しみ 悲しみ 形勢 が | **N3** 抽象 から に 始まる **(N1 arise from N3)**

◇ **N1** 時間 単位 抽象 組織 が | **N3** 時　分 時間 に から 始まる **(N1 begin at N3)**

N1 : ◇ **0** ◇三十一日 ◆学校

N3 : ◇ **0** ◆三十一日 ◇学校

Figure 5: Screen shot of verb sense diambiguation application

The left frame of the application displays the transcript of a CHJ conversation. NPs in the transcript are enclosed in square brackets, and main verbs are underlined and hyperlinked. Clicking on a main verb in the transcript (left frame) brings up a verb sense menu for that verb in the right frame.

For example, in Figure 5, the verb sense menu for a token of the verb 始まる *hazimaru* "begin" (fourth utterance in the left frame) has been brought up in the right frame. The four **GT** senses of *hazimaru* are listed as menu choices. Each sense is assigned a unique subcategorization frame, including the case roles (N1, N2, etc.; cf. Table 2). The subcategorized-for semantic categories are also underlined and hyperlinked to a diagram of the complete **GT** ontology (cf. Figure 2), so that the coders can see examples of each category and how that category fits into the broader semantic framework. Finally, an English gloss of each verb sense (from the **GT** transfer component) is given at the end of each subcategorization frame.

Once a coder selects the correct verb sense for the verb token in question (in the case of Figure 5, it is the first sense listed), the coder then selects that verb's NP complements (case-role fillers), if any, from within the same utterance. For example, in Figure 5, the coder has selected the first verb sense for 始まる *hazimaru* "begin", which subcategorizes for the NP arguments N1 (subject) and N3 (start time). Separate menu forms are displayed for both N1 and N3, with all NPs in the utterance listed as possible fillers. In this case the coder selected 学校 *gakkou* "school" as the subject and 三十一日 *sanjuuichinichi* "the 31st" as the start time. If no NP in the utterance fills a given role, the zero option is selected (this is the default choice). Finally, the coder clicks '`Submit`' and then moves on to the next verb in the transcript (left frame).

Acknowledgments

This project is supported by U.S. National Science Foundation grant BCS-0002646, and by a dissertation grant in Japanese studies to the first author from Stanford University's Institute for International Studies.

References

Y. Akiba, M. Ishii, H. Almuallim, and S. Kaneda. 1995. Learning English verb selection rules from hand-made rules and translation examples. *Sixth International Conference on Theoretical and Methodological Issues in Machine Translation: TMI-95*, pp. 206–220, Leuven, July.

T. Baldwin and H. Tanaka. 1999. Argument status in Japanese verb sense disambiguation. *TMI-99*, pp. 196–206, Chester, UK, August.

T. Baldwin, F. Bond, and B. Hutchinson. 1999. A valency dictionary architecture for machine translation. *TMI-99*, pp. 207–217, Chester, UK, August.

S. Bird and M. Liberman. 1998. Towards a formal framework for linguistic annotations. *ICSLP'98*, Sydney.

F. Bond and S. Shirai. 1997. Practical and efficient organization of a large valency dictionary. *NLPRS-97 Workshop on Multilingual Information Processing*, Phuket. (handout).

J. Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

M. Core, M. Ishizaki, J. Moore, C. Nakatani, N. Reithinger, D. Traum, and S. Tutiya. 1999. Report of the third workshop of the Discourse Resource Initiative. Technical Report CC-TR-99-1, Chiba Corpus Project, Chiba U.

L. Dybkjær, N. Bernson, H. Dybkjær, D. McKelvie, and A. Mengel. 1998. The MATE markup framework. MATE Deliverable D1.2, LE Telematics Project LE4–8370, Denmark. `http://mate.nis.sdu.dk/`.

EDR. 1996. EDR Electronic Dictionary Version 1.5. Japan Electronic Dictionary Research Institute, Ltd. `http://www.iijnet.or.jp/edr/`.

M. Hamanishi and S. Ono, editors. 1990. *Ruigo Kokugo Jiten [Japanese Synonym Dictionary]*. Kadokawa Shoten, Tokyo.

N. Ide. 1998. Encoding linguistic corpora. *Proceedings of the Sixth Workshop on Very Large Corpora*, pp. 9–17, Montreal.

S. Ikehara, S. Shirai, and K. Ogura. 1994. Criteria for evaluating the linguistic quality of Japanese to English machine translations. *Journal of Japanese Society for Artificial Intelligence*, 9(4):569–579. (In Japanese).

S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi. 1997. *Goi-Taikei — A Japanese Lexicon*. Iwanami Shoten, Tokyo. 5 volumes. `http://www.kecl.ntt.co.jp/icl/mtg/resources/GoiTaikei`.

IPA. 1987. *IPAL Lexicon of Basic Verbs*. Software Technology Center, Information-Technology Promotion Agency (IPA), Tokyo. (In Japanese).

IPA. 1990. *IPAL Lexicon of Basic Adjectives*. IPA, Tokyo. (In Japanese).

IPA. 1996. *IPAL Lexicon of Basic Nouns*. IPA, Tokyo. (In Japanese).

A. Kilgarriff and J. Rosenzweig. 1999. SENSEVAL: Report and results. *NLPRS'99*, pp. 362–367, Beijing, November.

A. Kilgarriff. 1998. Gold standard datasets for evaluating word sense disambiguation programs. *Computer Speech and Language*, 12(4):453–472.

LDC. 1996. Callhome Japanese corpus. Linguistic Data Consortium, University of Pennsylvania. `http://www.ldc.upenn.edu`.

H. Nakaiwa and K. Seki. 1999. Automatic addition of verbal semantic attributes to a Japanese-to-English valency transfer dictionary. *TMI-99*, pp. 185–195, Chester, UK.

M. Oku. 1996. Analyzing Japanese double-subject construction having an adjective predicate. *COLING-96*, pp. 865–870.

S. Shirai, S. Ikehara, A. Yokoo, and H. Inoue. 1995. The quantity of valency pattern pairs required for Japanese to English machine translation and their compilation. *NLPRS-95*, pp. 443–448, Seoul.

S. Yokoyama and T. Ochiai. 1999. Aimai-na sūryōshi-o fukumu meishiku-no kaisekihō [a method for analysing noun phrases with ambiguous quantifiers]. *ANLP-99*, pp. 550–553. (In Japanese).

Exploring automatic word sense disambiguation with decision lists and the Web

Eneko Agirre
IxA NLP group.
649 pk.
Donostia, Basque Country, E-20.080
eneko@si.ehu.es

David Martínez
IxA NLP group.
649 pk.
Donostia, Basque Country, E-20.080
jibmaird@si.ehu.es

Abstract

The most effective paradigm for word sense disambiguation, supervised learning, seems to be stuck because of the knowledge acquisition bottleneck. In this paper we take an in-depth study of the performance of decision lists on two publicly available corpora and an additional corpus automatically acquired from the Web, using the fine-grained highly polysemous senses in WordNet. Decision lists are shown a versatile state-of-the-art technique. The experiments reveal, among other facts, that SemCor can be an acceptable (0.7 precision for polysemous words) starting point for an all-words system. The results on the DSO corpus show that for some highly polysemous words 0.7 precision seems to be the current state-of-the-art limit. On the other hand, independently constructed hand-tagged corpora are not mutually useful, and a corpus automatically acquired from the Web is shown to fail.

Introduction

Recent trends in word sense disambiguation (Ide & Veronis, 1998) show that the most effective paradigm for word sense disambiguation is that of supervised learning. Nevertheless, current literature has not shown that supervised methods can scale up to disambiguate all words in a text into reference (possibly fine-grained) word senses. Possible causes of this failure are:

1. Problem is wrongly defined: tagging with word senses is hopeless. We will not tackle this issue here (see discussion in the Senseval e-mail list – senseval-discuss@sharp.co.uk).
2. Most tagging exercises use idiosyncratic word senses (e.g. ad-hoc built senses, translations, thesaurus, homographs, ...) instead of widely recognized semantic lexical resources (ontologies like Sensus, Cyc, EDR, WordNet, EuroWordNet, etc., or machine-readable dictionaries like OALDC, Webster's, LDOCE, etc.) which usually have fine-grained sense differences. We chose to work with WordNet (Miller et al. 1990).
3. Unavailability of training data: current hand-tagged corpora seem not to be enough for state-of-the-art systems. We test how far can we go with existing hand-tagged corpora like SemCor (Miller et al. 1993) and the DSO corpus (Ng and Lee, 1996), which have been tagged with word senses from WordNet. Besides we test an algorithm that automatically acquires training examples from the Web (Mihalcea & Moldovan, 1999).

In this paper we focus on one of the most successful algorithms to date (Yarowsky 1994), as attested in the Senseval competition (Kilgarriff & Palmer, 2000). We will evaluate it on both SemCor and DSO corpora, and will try to test how far could we go with such big corpora. Besides, the usefulness of hand tagging using WordNet senses will be tested, training on one corpus and testing in the other. This will allow us to compare hand tagged data with automatically acquired data.

If new ways out of the acquisition bottleneck are to be explored, previous questions about supervised algorithms should be answered: how much data is needed, how much noise can they accept, can they be ported from one corpus to another, can they deal with really fine sense distinctions, performance etc. There are few in-depth analysis of algorithms, and precision figures are usually the only features available. We designed a series of experiments in order to shed light on the above questions.

In short, we try to test how far can we go with current hand-tagged corpora, and explore whether other means can be devised to complement hand-tagged corpora. We first present decision lists and the features used, followed by the method to derive data from the Web and the design of the experiments. The experiments are organized in three sections: experiments on SemCor and DSO, cross-corpora experiments, and tagging SemCor using the Web data for training. Finally some conclusions are drawn.

1 Decision lists and the features used

Decision lists (DL) as defined in (Yarowsky, 1994) are simple means to solve ambiguity problems. They have been successfully applied to accent restoration, word sense disambiguation and homograph disambiguation (Yarowsky, 1994; 1995; 1996). It was one of the most successful systems on the Senseval word sense disambiguation competition (Kilgarriff and Palmer, 2000).

The training data is processed to extract the features, which are weighted with a log-likelihood measure. The list of all features ordered by the log-likelihood values constitutes the decision list. We adapted the original formula in order to accommodate ambiguities higher than two:

$$weight(sense_i, feature_k) = Log(\frac{\Pr(sense_i \mid feature_k)}{\sum_{j \neq i} \Pr(sense_j \mid feature_k)})$$

Features with 0 or negative values were are not inserted in the decision list.

When testing, the decision list is checked in order and the feature with highest weight that is present in the test sentence selects the winning word sense. An example is shown below.

The probabilities have been estimated using the maximum likelihood estimate, smoothed using a simple method: when the denominator in the formula is 0 we replace it with 0.1**.**

We analyzed several **features** already mentioned in the literature (Yarowsky, 1994; Ng, 1997; Leacock et al. 1998), and new features like the word sense or semantic field of the words around the target which are available in SemCor. Different sets of features have been created to test the influence of each feature type in the results: a basic set of features (section 4), several extensions (section 4.2).

The example below shows three senses of the noun *interest*, an example, and some of the features for the decision lists of *interest* that appear in the example shown.

Sense 1: interest, involvement => curiosity, wonder
Sense 2: interest, interestingness => power, powerfulness, potency
Sense 3: sake, interest => benefit, welfare

.... considering the widespread interest in the election ...

2.99 '#3 lem_50w win 2 2'
1.54 '#2 big_wf_-1 interest in 14 17'
1.25 '#2 big_lem_-1 in 14 18'

We see that the feature which gets the highest weight (2.99) is "lem_50w win" (the lemma *win* occurring in a 50-word window). The lemma *win* shows up twice near *interest* in the training corpus and always indicates the sense #3. The next best feature is " big_wf_-1 interest in" (the bigram "interest in") which in 14 of his 17 apparitions indicates sense #2 of *interest*. Other features follow. The interested reader can refer to the papers where the original features are described.

2 Deriving training data from the Web

In order to derive automatically training data from the Web, we implemented the method in (Mihalcea & Moldovan, 1999). The information in WordNet (e.g. monosemous synonyms and glosses) is used to construct queries that are later fed into a web search engine like Altavista. Four procedures can be used consecutively, in decreasing order of precision, but with increasing amounts of examples retrieved. Mihalcea and Moldovan evaluated by hand 1080 retrieved instances of 120 word senses, and attested that 91% were correct. The method was not used to train a word sense disambiguation system.

In order to train our decision lists, we automatically retrieved around 100 documents per word sense. The html documents were converted into ASCII texts, and segmented into paragraphs and sentences. We only used the sentence around the target to train the decision lists. As the gloss or synonyms were used to retrieve the text, we had to replace those with the target word.

The example below shows two senses of *church*, and two samples for each. For the first sense, part of the gloss, *group of Christians* was used to retrieve the example shown. For the second sense, the monosemous synonyms *church building* was used.

'church1' => GLOSS 'a group of Christians'
Why is one >> church << satisfied and the other oppressed ? :

'church2' => MONOSEMOUS SYNONYM 'church building'
The result was a congregation formed at that place, and a >> church << erected . :

Several improvements can be made to the process, like using part-of-speech tagging and morphological processing to ensure that the replacement is correctly made, discarding suspicious documents (e.g. indexes, too long or too short) etc. Besides (Leacock et al., 1998) and (Agirre et al., 2000) propose alternative strategies to construct the queries. We chose to evaluate the method as it stood first, leaving the improvements for the future.

3 Design of the experiments

The experiments were targeted at three different corpora. **SemCor** (Miller et al., 1993) is a subset of the Brown corpus with a number of texts comprising about 200.000 words in which all content words have been manually tagged with senses from WordNet (Miller et al. 1990). It has been produced by the same team that created WordNet. As it provides training data for all words in the texts, it allows for all-word evaluation, that is, to measure the performance all the words in a given running text. The **DSO corpus** (Ng and Lee, 1996) was differently designed. 191 polysemous words (nouns and verbs) and an average of 1000 sentences per word were selected from the Wall Street Journal and Brown corpus. In the 192.000 sentences only the target word was hand-tagged with WordNet senses. Both corpora are publicly available. Finally, a **Web corpus** (cf. section 2) was automatically acquired, comprising around 100 examples per word sense.

For the experiments, we decided to focus on a few content words, selected using the following criteria: 1) the frequency, according to the number of training examples in SemCor, 2) the ambiguity level 3) the skew of the most frequent sense in SemCor, that is, whether one sense dominates.

The two first criteria are interrelated (frequent words tend to be highly ambiguous), but there are exceptions. The third criterion seems to be independent, but high skew is sometimes related to low ambiguity. We could not find all 8 combinations for all parts of speech and the following samples were selected (cf. Table 1): 2 adjectives, 2 adverbs, 8 nouns and 7 verbs. These 19 words form the **test set A**.

The DSO corpus does not contain adjectives or adverbs, and focuses on high frequency words. Only 5 nouns and 3 verbs from Set A were present in the DSO corpus, forming **Set B** of test words.

In addition, **4 files from SemCor** previously used in the literature (Agirre & Rigau, 1996) were selected, and all the content words in the file were disambiguated (cf. section 4.7).

The measures we use are precision, recall and coverage, all ranging from 0 to 1. Given N, number of test instances, A, number of instances which have been tagged, and C, number of instances which have been correctly tagged; precision = C/A, recall = C/N and coverage =A/ N In fact, we used a modified measure of precision, equivalent to choosing at random in ties.

The experiments are organized as follows:

- Evaluate decision lists on SemCor and DSO separately, focusing on baseline features, other features, local vs. topical features, learning curve, noise, overall in SemCor and overall in DSO (section 4). All experiments were performed using 10-fold cross-validation.
- Evaluate cross-corpora tagging. Train on DSO and tag SemCor and vice versa (section 5).
- Evaluate the Web corpus. Train on Web-acquired texts and tag SemCor (section 6).

Because of length limitations, it is not possible to show all the data, refer to (Agirre & Martinez, 2000) for more comprehensive results.

4 Results on SemCor and DSO data

We first defined an initial set of features and compared the results with the random baseline (Rand) and the most frequent sense baseline (MFS). The basic combination of features comprises word-form bigrams and trigrams, part of speech bigrams and trigrams, a bag with the word-forms in a window spanning 4 words left and right, and a bag with the word forms in the sentence.

The results for SemCor and DSO are shown in Table 1. We want to point out the following:

- **The number of examples per word sense is very low for SemCor** (around 11 for the words in Set B), while DSO has substantially more training data (around 66 in set B). Several word senses occur neither in SemCor nor in DSO.
- **The random baseline** attains 0.17 precision for Set A, and 0.10 precision for Set B.
- **The MFS baseline** is higher for the DSO corpus (0.59 for Set B) than for the SemCor corpus (0.50 for Set B). This rather high discrepancy can be due to tagging disagreement, as will be commented on section 5.
- Overall, **decision lists significantly outperform the two baselines** in both corpora: for set B 0.60 vs. 0.50 in SemCor, and 0.70 vs. 0.59 on DSO, and for Set A 0.70 vs. 0.61 on SemCor. For a few words the decision lists trained on SemCor are not able to beat MFS (results in bold), but in DSO decision lists overcome in all words. **The scarce data in SemCor seems enough to get some basic results. The larger amount of data in DSO warrants a better performance, but limited to 0.70 precision.**
- **The coverage in SemCor does not reach 1.0**, because some decisions are rejected when the log

				SemCor				DSO			
Word	PoS	Senses	Rand	# Examples	Ex. Per sense	MFS	DL	# Examples	Ex. Per senses	MFS	DL
All	A	2	.50	211	105.50	**.99**	**.99**/1.0				
Long	A	10	.10	193	19.30	.53	**.63**/.99				
Most	B	3	.33	238	79.33	.74	**.78**/1.0				
Only	B	7	.14	499	71.29	.51	**.69**/1.0				
Account	N	10	.10	27	2.70	.44	**.57**/.85				
Age	N	5	.20	104	20.80	.72	**.76**/1.0	491	98.20	.62	**.73**/1.0
Church	N	3	.33	128	42.67	.41	**.69**/1.0	370	123.33	.62	**.71**/1.0
Duty	N	3	.33	25	8.33	.32	**.61**/.92				
Head	N	30	.03	179	5.97	.78	**.88**/1.0	866	28.87	.40	**.79**/1.0
Interest	N	7	.14	140	20.00	.41	**.62**/.97	1479	211.29	.46	**.62**/1.0
Member	N	5	.20	74	14.80	**.91**	**.91**/1.0	1430	286.00	.74	**.79**/1.0
People	N	4	.25	282	70.50	**.90**	**.90**/1.0				
Die	V	11	.09	74	6.73	**.97**	**.97**/.99				
Fall	V	32	.03	52	1.63	.13	**.34**/.71	1408	44.00	.75	**.80**/1.0
Give	V	45	.02	372	8.27	.22	**.34**/.78	1262	28.04	.75	**.77**/1.0
Include	V	4	.25	144	36.00	**.72**	.70/.99				
Know	V	11	.09	514	46.73	.59	**.61**/1.0	1441	131.0	.36	**.46**/.98
Seek	V	5	.20	46	9.20	.48	**.62**/.89				
Understand	V	5	.20	84	16.80	**.77**	**.77**/1.0				
Avg. A		5.82	.31	202.00	34.71	.77	**.82**/1.0				
Avg. B		5.71	.20	368.50	64.54	.58	**.72**/1.0				
Set A Avg. N		9.49	.19	119.88	12.63	.69	**.80**/.99				
Avg. V		20.29	.10	183.71	9.05	.51	**.58**/.92				
Overall		12.33	.17	178.21	14.45	.61	**.70**/.97				
Avg. N		10.00	.16	125.00	12.50	.63	**.77**/.99	927.20	92.72	.56	**.72**/1.0
Set B Avg. V		29.33	.06	312.67	10.66	.42	**.49**/.90	137.33	46.72	.61	**.67**/.99
Overall		17.25	.10	195.38	11.33	.50	**.60**/.94	1093.38	63.38	.59	**.70**/1.0

Table 1: Data for each word and results for baselines and basic set of features.

likelihood is below 0. On the contrary, the richer data in DSO enables 1.0 coverage.

Regarding the execution time, Table 3 shows training and testing times for each word in SemCor. Training the 19 words in set A takes around 2 hours and 30 minutes, and is linear to the number of training examples, around 2.85 seconds per example. Most of the training time is spent processing the text files and extracting all the features, which includes complex window processing. Once the features have been extracted, training time is negligible, as is the test time (around 2 seconds for all instances of a word). Time was measured on CPU total time on a Sun Sparc 10 (512 MB of memory at 360 MHz).

4.1 Results in SemCor according to the kind of words: skew of MFS counts

We plotted the precision attained in SemCor for each word, according to certain features. Figure 1 shows the precision according to the frequency of each word, measured in number of occurrences in SemCor. Figure 2 shows the precision of each word plotted according to the number of senses. Finally, Figure 3 orders the words according to the degree of dominance of the most frequent sense. The figures show the precision of decision lists (DL), but also plot the difference of performance according to two baselines, random (DL-Rand) and MFS (DL-MFS). These last figures are close to 0 whenever decision lists attain results similar to those of the baselines. We observed the following:

- Contrary to expectations, **frequency and ambiguity do not affect precision** (Figures 1 and 2). This can be explained by interrelation between ambiguity and frequency. Low ambiguity words may seem easier to disambiguate, but they tend to occur less, and SemCor provides less data. On the contrary, highly ambiguous words occur more frequently, and have more training data.
- **Skew does affect precision.** Words with high skew obtain better results, but decision lists outperform MFS mostly on words with low skew.

Overall decision lists perform very well (related to MFS) even with words with very few examples (“duty”, 25 or “account”, 27) or highly ambiguous words.

4.2 Features: basic features are enough

Our next step was to test other alternative features. We analyzed different window sizes (20 words, 50 words, the surrounding sentences), and used word lemmas, synsets and semantic fields. We also tried mapping the fine-grained part of speech distinctions in SemCor to a more general

Word	Base	±1sent	±20w	±50w	Lemmas	Synsets	Semantic	General
	Features						Fields	PoS
Avg. Adj.	.82/1.0	.79/1.0	.82/1.0	.81/1.0	.81/1.0	.82/1.0	**.84**/1.0	.82/1.0
Avg. Adv.	**.72**/1.0	.68/1.0	.68/1.0	.70/1.0	.69/1.0	**.72**/1.0	**.72**/1.0	.69/1.0
Avg. Nouns	.80/.99	.79/1.0	.80/1.0	.79/1.0	**.81**/1.0	.80/.99	.80/1.0	.80/.99
Avg. Verbs	.58/.92	.54/.98	.55/.97	.53/.99	.56/.95	.57/.94	.58/.93	**.59**/.89
Overall	.70/.97	.67/.99	.68/.99	.68/1.0	.69/.98	.70/.98	**.71**/.97	.70/.95

Table 2: Results with different sets of features.

set (nouns, verbs, adj., adv., others), and combinations of PoS and word form trigrams. Most of these features are only available in SemCor: context windows larger than sentence, synsets/semantic files of the open class words in the context.

The results are illustrated in Table 2 (winning combinations in bold). We clearly see that there is no significant loss or gain of accuracy for the different feature sets. **The use of wide windows sometimes introduces noise and the precision drops slightly**. At this point, we cannot be conclusive, as SemCor files mix text from different sources without any marking.

Including lemma or synset information does not improve the results, but taking into account the semantic files for the words in context improves one point overall. If we study each word, there is little variation, except for church: the basic precision (0.69) is significantly improved if we take into account semantic file or synset information, but specially if lemmas are contemplated (0.78 precision).

Besides, including all kind of dependent features does not degrade the performance significantly, showing that decision lists are resistant to spurious features.

4.3 Local vs. Topical: local for best prec., combined for best cov.

We also analyzed the performance of topical features versus local features. We consider as local bigrams and trigrams (PoS tags and word-forms), and as topical all the word-forms in the sentence plus a 4 word-form window around the target. The results are shown in Table 4.

The part of speech of the target influences the results: in SemCor, we can observe that while the topical context performed well for nouns, the accuracy dropped for the categories. These results are consistent with those obtained by (Gale et al. 1993) and (Leacock et al. 1998), which show that topical context works better for nouns. However, the results in the DSO are in clear contradiction with those from SemCor: local features seem to perform better for all parts of speech. It is hard to explain the reasons for this contradiction, but it can be related to the amount of data in DSO.

Word		Senses	Examples	Ex. Per	Testing	Training
				sense	time	time
					(secs)	(secs)
Set A	Avg. A	5.82	202.00	34.71	2.00	728.20
	Avg. B	5.71	368.50	64.54	3.80	997.65
	Avg. N	9.49	119.88	12.63	1.04	328.91
	Avg. V	20.29	183.71	9.05	1.66	510.63

Table 3: Execution time for the words in SemCor.

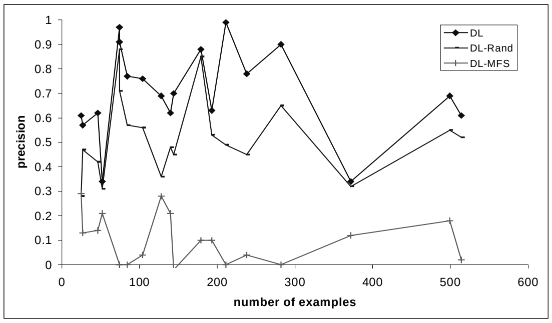


Figure 1: Results of DL and baselines according to frequency.

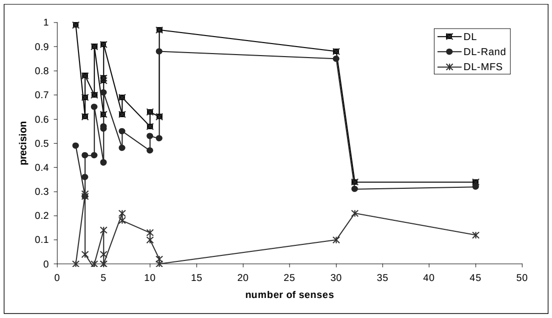


Figure 2: Results according to ambiguity**.**

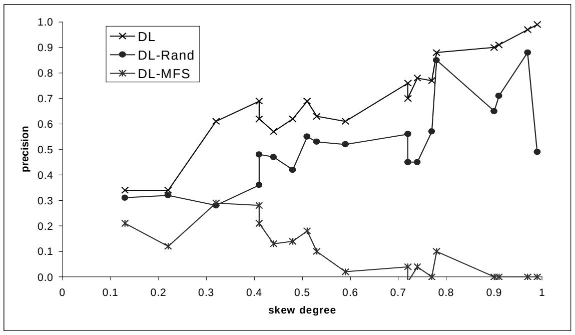


Figure 3: Results according to skew.

The combination all features attains lower precision in average than the local features alone, but this is compensated by a higher coverage, and overall the recall is very similar in both corpora

4.4 Learning curve: examples in DSO enough

We tested the performance of decision lists with different amounts of training data. We retained increasing amounts of the examples available for each word: 10% of all examples in the corpus, 20%, 40%, 60%, 80% and 100%. We performed 10 rounds for each percentage of training data, choosing different slices of data for training and testing. Figures 4 and 5 show the number of training examples and recall obtained for each percentage of training data in SemCor and DSO respectively. Recall was chosen in order to compensate for differences in both precision and coverage, that is, recall reflects both decreases in coverage and precision at the same time.

The improvement for nouns in SemCor seem to stabilize, but the higher amount of examples in DSO show that the performance can still grow up to a standstill. The verbs show a steady increase in SemCor, confirmed by the DSO data, which seems to stop at 80% of the data.

4.5 Noise: more data better for noise

In order to analyze the effect of noise in the training data, we introduced some random tags in part of the examples. We created 4 new samples for training, with varying degrees of noise: 10% of the examples with random tags, %20, %30 and 40%.

Figures 6 and 7 show the recall data for SemCor and DSO. The decrease in recall is steady for both nouns and verbs in SemCor, but it is rather brusque in DSO. **This could mean that when more data is available, the system is more robust to noise**: the performance is hardly affected by %10, 20% and 30% of noise.

4.6 Coarse Senses: results reach .83 prec.

It has been argued that the fine-grainedness of the sense distinctions in SemCor makes the task more difficult than necessary. WordNet allows to make sense distinctions at the semantic file level, that is, the word senses that belong to the same semantic file can be taken as a single sense (Agirre & Rigau, 1996). We call the level of fine-grained original senses *the synset level*, and the coarser senses form *the semantic file level*.

In case any work finds these coarser senses useful, we trained the decision lists with them both in SemCor and DSO. The results are shown in Table 5 for the words in Set B. At this level the results on both corpora reach 83% of precision.

4.7 Overall Semcor: .68 prec. for all-word

In order to evaluate the expected performance of decision lists trained on SemCor, we selected four

	SemCor			DSO		
PoS	Local	Topical	Comb.	Local	Topical	Comb.
A	**.84**/.99	.81/.89	.82/**1.0**			
B	**.74**/1.0	.64/.96	.72/**1.0**			
N	.78/.96	**.81**/.87	.80/**.99**	**.75**/.97	.71/.98	.72/**1.0**
V	**.61**/.84	.57/.72	.58/**.92**	**.70**/.96	.66/.91	.67/**.99**
Ov.	**.72**/.93	.68/***.84***	.70/**.97**	**.73**/.96	.69/.95	.70/**1.0**

Table 4: Local context Vs Topical context.

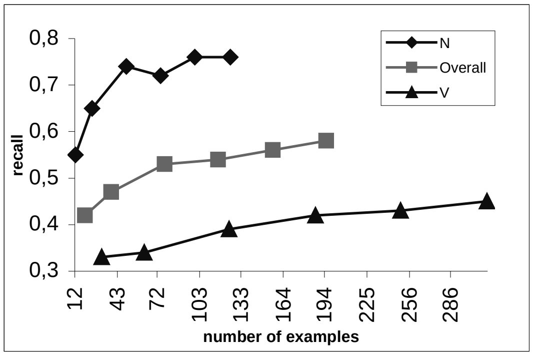


Figure 4: Learning curve in SemCor.

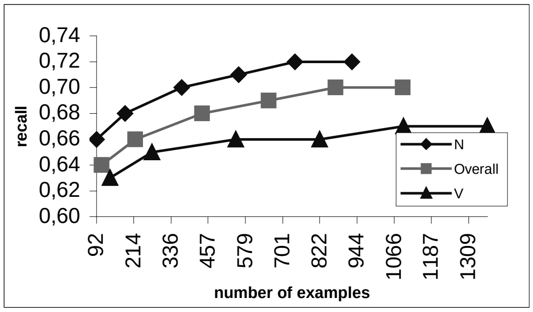


Figure 5: learning curve in DSO.

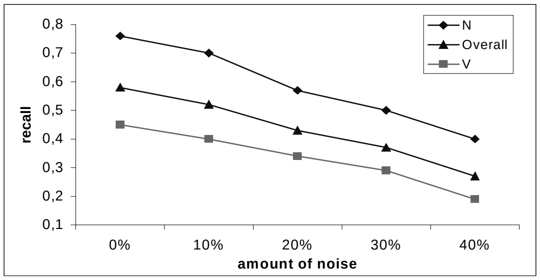


Figure 6: Results with noise in SemCor.

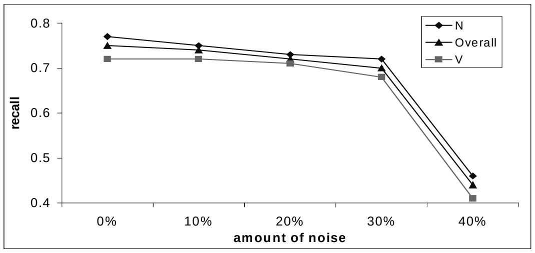


Figure 7: Results with noise in DSO.

files previously used in the literature (Agirre & Rigau, 1996) and all the content words in the files were disambiguated. For each file, the decision lists were trained with the rest of SemCor.

Table 6 shows the results. Surprisingly, decision lists attain a very similar performance in all four files (random and most frequent baselines also show the same behaviour). As SemCor is a balanced corpus, it seems reasonable to say that 68% precision can be expected if any running text is disambiguated using decision lists trained on SemCor. The fact that the results are similar for texts from different sources (journalistic, humor, science) and that similar results can be expected for words with varying degrees of ambiguity and frequency (cf. section 4.1), seems to confirm that the training data in SemCor allows to **expect for a similar precision across all kinds of words and texts**, except for highly skewed words, where we can expect better performance than average.

4.8 Overall DSO: state-of-the-art results

In order to compare decision lists with other state of the art algorithms we tagged all 191 words in the DSO corpus. The results in (Ng, 1997) only tag two subsets of all the data, but (Escudero et al. 2000a) implement both Ng's example-based (EB) approach and a Naive-Bayes (NB) system and test it on all 191 words. The same test set is also used in (Escudero et al. 2000b) which presents a boosting approach to word sense disambiguation. The features they use are similar to ours, but not exactly. The precision obtained, summarized on Table 7 show that **decision lists provide state-of-the-art** performance. Decision list attained 0.99 coverage.

5 Cross-tagging: hand taggers need to be coordinated

We wanted to check what would be the performance of the decision lists training on one corpus and tagging the other. The DSO and SemCor corpora do not use exactly the same word sense system, as the former uses WordNet version 1.5 and the later WordNet version 1.6. We were able to easily map the senses form one to the other for all the words in Set B. We did not try to map the word senses that did not occur in any one of the corpora.

			SemCor		DSO	
POS	# Syns	# SFs	Synset	SF	Synset	SF
N	50	29	.77/.99	.78/.00	.72/1.0	.76/1.0
V	88	19	.51/.90	.87/.96	.67/.99	.91/1.0
Ov.	138	48	.62/.94	.83/.98	.70/1.0	.83/1.0

Table 5: Results disambiguating fine (synset) vs. coarse (SF) senses.

File	POS	# Senses	# Examples	Rand	MFS	DL
br-a01		6.60	792	.26	.63	.68/.95
br-b20		6.86	756	.24	.64	.66/.95
br-j09		6.04	723	.24	.64	.69/.95
br-r05		7.26	839	.24	.63	.68/.92
average	A	5.49	122.00	.28	.71	.71/.92
	B	3.76	48.50	.34	.72	.80/.97
	N	4.87	366.75	.28	.66	.69/.94
	V	10.73	240.25	.16	.54	.61/.95
	Ov.	6.71	777.50	.25	.63	.68/.94

Table 6: Overall results in SemCor.

PoS	MFS	EB	NB	Boosting	Decision Lists
N	.59/1.0	.69	.68	.71	**.72**/.99
V	.53/1.0	.65	.65	.67	**.68**/.98
Ov	.56/1.0	.67	.67	**.70**	**.70**/.99

Table 7: Overall results in DSO.

A previous study (Ng et al. 1999) has used the fact that some sentences of the DSO corpus are also included in SemCor in order to study the agreement between the tags in both corpora. They showed that the hand-taggers of the DSO and SemCor teams only agree 57% of the time. This is a rather low figure, which explains why the results for one corpus or the other differ, e.g. the differences on the MFS results (see Table 1).

Considering this low agreement, we were not expecting good results on this cross-tagging experiment. The results shown in Table 8 confirmed our expectations, as the precision is greatly reduced (approximately one third in both corpora, but more than a half in the case of verbs). **Teams of hand-taggers need to be coordinated in order to produce results that are interchangeable**.

6 Results on Web data: disappointing

We used the Web data to train the decision lists (with the basic feature set) and tag the SemCor examples. Only nouns and verbs were processed, as the method would not work with adjectives and adverbs. Table 9 shows the number of examples retrieved for the target words, the random baseline and the precision attained. Only a few words get better than random results (in bold), and for *account* the error rate reaches 100%.

These extremely low results clearly contradict the optimism in (Mihalcea & Moldovan, 1999), where a sample of the retrieved examples was found to be 90% correct. One possible explanation of this apparent disagreement could be that the acquired examples, being correct on themselves, provide systematically misleading features. Besides, all word senses are trained with

Word	PoS	# Training Examples (in SemCor)	Cross MFS (in DSO)	Cross Prec./Cov. (in DSO)	Original Prec/Cov (in DSO)	# Training Examples (in DSO)	Cross MFS (SemCor)	Cross Prec./Cov. (SemCor)	Original Prec/Cov (SemCor)
Age	N	104	.62	.67/.97	.76/1.0	491	.72	.63/1.0	.73/1.0
Church	N	128	.62	.68/.99	.69/1.0	370	.47	.78/1.0	.71/1.0
Head	N	179	.40	.40/.97	.88/1.0	866	.03	.77/1.0	.79/1.0
Interest	N	140	.18	.37/.90	.62/.97	1479	.10	.35/.99	.62/1.0
Member	N	74	.74	.74/.97	.91/1.0	1430	.91	.84/1.0	.79/1.0
Fall	V	52	.01	.06/.54	.34/.71	1408	.04	.32/.96	.80/1.0
Give	V	372	.01	.16/.72	.34/.78	1262	.09	.15/1.0	.77/1.0
Know	V	514	.27	.32/1.0	.61/1.0	1441	.14	.44/.98	.46/.98
N		125.00	.48	.55/.95	.77/.99	927.20	.35	.66/1.0	.72/1.0
V		312.67	.10	.21/.76	.51/.90	137.33	.11	.32/.99	.67/.99
Overall		195.38	.30	.41/.86	.62/.94	1093.38	.21	.46/.99	.70/1.0

Table 8: Cross tagging the corpora.

equal number of examples, whichever their frequency in Semcor (e.g. word senses not appearing in SemCor also get 100 examples for training), and this could also mislead the algorithm.Further work is needed to analyze the source of the errors, and devise ways to overcome these worrying results.

Word	PoS	# Examples	Rand.	DL on SemCor
Account	N	1175	.10	.00/.85
Age	N	630	.20	**.29/.97**
Church	N	386	.33	**.46/.98**
Duty	N	449	.33	**.35/1.0**
Head	N	3636	.03	.04/.44
Interest	N	1043	.14	**.25/.88**
Member	N	696	.20	.16/.86
People	N	591	.25	.16/.95
Die	V	1615	.09	.04/.93
Include	V	577	.25	.11/.99
Know	V	1423	.09	.07/.64
Seek	V	714	.20	**.49/.98**
Understand	V	780	.20	.12/.92

Table 9: Results on Web data.

7 Conclusions and further work

This paper tries to tackle several questions regarding decision lists and supervised algorithms in general, in the context of a word senses based on a widely used lexical resource like WordNet. The conclusions can be summarized according to the issues involved as follows:

- **Decision lists:** this paper shows that decision lists provide state-of-the-art results with simple and very fast means. It is easy to include features, and they are robust enough when faced with spurious features. They are able to learn with low amounts of data.
- **Features:** the basic set of features is enough. Larger contexts than the sentence do not provide much information, and introduce noise. Including lemmas, synsets or semantic files does not significantly alter the results. Using a simplified set of PoS tags (only 5 tags) does not degrade performance. Local features, i.e. collocations, are the strongest kind of features, but topical features enable to extend the coverage.
- **Kinds of words:** the highest results can be expected for words with a dominating word sense. Nouns attain better performance with local features when enough data is provided. Individual words exhibit distinct behavior regarding to the feature sets.
- **SemCor** has been cited as having scarce data to train supervised learning algorithms (Miller et al., 1994). *Church,* for instance, occurs 128 times, but *duty* only 25 times and *account* 27. We found out that SemCor nevertheless provides enough data to perform some basic general disambiguation, at 0.68 precision on any general running text. The performance on different words is surprisingly similar, as ambiguity and number of examples are balanced in this corpus. The learning curve indicates that the data available for nouns could be close to being sufficient, but verbs have little available data in SemCor.
- **DSO** provides large amounts of data for specific words, allowing for improved precision. It is nevertheless stuck at 0.70 precision, too low to be useful at practical tasks. The learning curve suggests that an upper bound has been reached for systems trained on WordNet word senses and hand-tagged data. This figures contrast with higher figures (around 90%) attained by Yarowsky on the Senseval competition (Kilgarriff & Palmer, 2000). The difference could be due to the special nature of the word senses defined for the Senseval competition.
- **Cross-corpora tagging**: the results are disappointing. Teams involved in hand-tagging need to coordinate with each other, at the risk of generating incompatible data.
- **Amount of data and noise**: SemCor is more affected by noise than DSO. It could mean that

higher amounts of data provide more robustness from noise.

- **Coarser word senses**: If decision lists are trained on coarser word senses inferred from WordNet itself, 80% precision can be attained for both SemCor and DSO.
- **Automatic data acquisition from the Web**: the preliminary results shown in this paper show that the acquired data is nearly useless.

The goal of the work reported here was to provide the foundations to open-up the acquisition bottleneck. In order to pursue this ambitious goal we explored key questions regarding the properties of a supervised algorithm, the upper bounds of manual tagging, and new ways to acquire more tagging material.

According to our results hand-tagged material is not enough to warrant useful word sense disambiguation on fine-grained reference word senses. On the other hand, contrary to current expectations, automatically acquisition of training material from the Web fails to provide enough support.

In the immediate future we plan to study the reasons for this failure and to devise ways to improve the quality of the automatically acquired material.

Acknowledgements

The work here presented received funds from projects OF319-99 (Government of Gipuzkoa), EX1998-30 (Basque Country Government) and 2FD1997-1503 (European Commission).

Bibliography

Agirre E. and Rigau G. *Word Sense Disambiguation using Conceptual Density*. Proceedings of COLING'96, 16-22. Copenhagen (Denmark). 1996.

Agirre, E., O. Ansa, E. Hovy and D. Martinez *Enriching very large ontologies using the WWW*. ECAI 2000, Workshop on Ontology Learning. Berlin, Germany. 2000.

Agirre, E. and D. Martinez. *Exploring automatic word sense disambiguation with decision lists and the Web. Internal report*. UPV-EHU. Donostia, Basque Country. 2000.

Escudero, G., L. Màrquez and G. Rigau. *Naive Bayes and Exemplar-Based approaches to Word Sense Disambiguation Revisited*. Proceedings of the 14th European Conference on Artificial Intelligence, ECAI 2000. 2000.

Escudero, G., L. Màrquez and G. Rigau. *Boosting Applied to Word Sense Disambiguation*. Proceedings of the 12th European Conference on Machine Learning, ECML 2000. Barcelona, Spain. 2000.

Gale, W., K. W. Church, and D. Yarowsky. *A Method for Disambiguating Word Senses in a Large Corpus*, Computers and the Humanities, 26, 415--439, 1993.

Ide, N. and J. Veronis. *Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art*. Computational Linguistics, 24(1), 1--40, 1998.

Kilgarriff, A. and M. Palmer. (eds). *Special issue on SENSEVAL*. Computer and the Humanities, 34 (1-2). 2000

Leacock, C., M. Chodorow, and G. A. Miller. *Using Corpus Statistics and WordNet Relations for Sense Identification.* Computational Linguistics, 24(1), 147--166, 1998.

Mihalcea, R. and I. Moldovan. *An Automatic Method for Generating Sense Tagged Corpora.* Proceedings of the 16th National Conference on Artificial Intelligence. AAAI Press, 1999.

Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. *Five Papers on WordNet*. Special Issue of International Journal of Lexicography, 3(4), 1990.

Miller, G. A., C. Leacock, R. Tengi, and R. T. Bunker, *A Semantic Concordance*. Proceedings of the ARPA Workshop on Human Language Technology, 1993.

Miller, G. A., M. Chodorow, S. Landes, C. Leacock and R. G. Thomas. *Using a Semantic Concordance for Sense Identification.* Proceedings of the ARPA. 1994.

Ng, H. T. and H. B. Lee. *Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-based Approach.* Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics. 1996.

Ng, H. T. *Exemplar-Based Word Sense Disambiguation: Some Recent Improvements.* Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, 1997.

Ng, H. T., C. Y. Lim and S. K. Foo. *A Case Study on Inter-Annotator Agreement for Word Sense Disambiguation.* Proceedings of the Siglex-ACL Workshop on Standarizing Lexical Resources. 1999.

Yarowsky, D. Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French', in Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 88--95. 1994.

Yarowsky, D. *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods.* Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics. Cambridge, MA, pp. 189-196, 1995.

Yarowsky, D. *Homograph Disambiguation in Text-to-speech Synthesis.* J Hirschburg, R. Sproat and J. Van Santen (eds.) Progress in Speech Synthesis, Springer-Vorlag, pp. 159-175. 1996.

Improving Natural Language Processing by Linguistic Document Annotation

Hideo Watanabe*, Katashi Nagao*, Michael C. McCord** and Arendse Bernth**

* IBM Research, Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato,
Kanagawa 242-8502, Japan
watanabe@trl.ibm.co.jp, nagao@trl.ibm.co.jp

** IBM T. J. Watson Research Center
Route 134, Yorktown Heights,
NY 10598, USA
mcmccord@us.ibm.com, arendse@us.ibm.com

Abstract

Natural language processing (NLP) programs are confronted with various difficulties in processing HTML and XML documents, and have the potential to produce better results if linguistic information is annotated in source texts. We have therefore developed the Linguistic Annotation Language (or LAL), which is an XML-compliant tag set for assisting natural language processing programs. It consists of linguistic information tags such as tags specifying word/phrasal boundaries, and task-dependent instruction tags such as tags defining the scope of translation for machine translation programs. We have also developed an LAL-annotation editor to facilitate users to annotate documents without seeing tags.

1 Introduction

The rapid expansion of the Internet has accelerated the proliferation of documents written in HTML and XML. Programs for performing natural language processing (or NLP) tasks such as keyword extraction, automatic text summarization, and machine translation have to be able to deal with such Internet documents. However, there are various obstacles that make it difficult for them to produce good results. It is true that NLP technologies are not perfect, but some of the difficulties result from problems in HTML. Further, in general, if linguistic information is added in a source text, it greatly helps NLP programs to produce a better result. Consider the following situations. When you use a search engine, you are often returned a list of thousands of documents matching your query. Most of the current search engines just use superficial information such as keywords. If search engines used richer linguistic information such as syntactic structures, they would be able to provide a more appropriate ranking of retrieved documents.

HTML Source
I used the h3 tag to emphasize ⟨h3⟩this part⟨/h3⟩.
Rendering Image
I used the h3 tag to emphasize **this part** .

Figure 1: An example of wrong usage of HTML tag

When you generate a summary of an HTML page by using an automatic summary generation program, a copyright notice is sometimes included in the summary text. Most of the current automatic summary programs simply select important sentences on the basis of surface clues such as keywords and sentence location in a document. As a result, they sometimes select a copyright notice located at the end of a document, since sentences located at the ends of documents tend to be important. This problem can be avoided if the main part of document is explicitly declared.

Further, when you use a Web page translation program, you sometimes see wrong translations. Most of them are generated by the incompleteness of MT technology, but some are generated by problems involving HTML and XML tag usage. For instance, writers often misuse tags to obtain certain stylistic effects. For instance, some writers use a heading tag to obtain large font and bold style, as shown in Fig. 1. Most machine translation (MT) engines change the translation logic when a sentence is a title, so this wrong use of heading tags sometimes causes a wrong translation result. However, the likelihood of this will decrease if a style sheet mechanism is widely accepted by Web authors in the future.

Another example of HTML/XML problems is the recognition of a sentence. There are many cases in which a sentence is terminated not by a period, but merely by a ⟨br⟩ tag, for instance, in an HTML table environment. As shown in Fig. 2, a writer

```
⟨table⟩
⟨tr⟩
⟨td⟩
⟨a href=”...”⟩Internet Shops⟨/a⟩⟨br⟩
⟨a href=”...”⟩Cool Sites⟨/a⟩⟨br⟩
⟨a href=”...”⟩What’s New!⟨/a⟩
⟨/td⟩
⟨/tr⟩
⟨/table⟩
```

Figure 2: An example of using ⟨br⟩ tags in a table

sometimes intends each line in a cell of a table to express a sentence, even if there is no punctuation at the end of the line. The MT program cannot tell whether each line is a sentence or whether these three lines form one sentence.

In general, it is very helpful for machine translation programs to know boundaries in many levels (such as sentence, phrases, and words) and to know word-to-word dependency relations. For instance, in the following example, ”St.” has two possible meanings: ”street” and ”saint.” Therefore, we cannot determine whether the following example consists of one or two sentences without parsing it.

> I went to New Ark St. Paul lived there
> in two years ago.

As another example, the following sentence is ambiguous so that there are two interpretations; one interpretation is that what he likes is people and the other interpretation is that what he likes is accommodating. If there are tags indicating the direct-object modifier of the word ”like,” then the correct interpretation is possible.

> He likes accommodating people.

As the above examples show, NLP applications do not achieve their full potential, on account of problems unrelated to the essential NLP processes. If tags expressing linguistic information are inserted into source documents, they help NLP programs recognize document and linguistic structures properly, allowing the programs to produce much better results. At the same time, it is true that NLP technologies are incomplete, but their deficiencies can sometimes be circumvented through the use of such tags. Therefore, this paper proposes a set of tags for helping NLP programs, called Linguistic Annotation Language (or LAL).

2 Linguistic Annotation Language

2.1 Design Principle

Linguistic Annotation Language (or LAL) is an XML-compliant tag set. It was designed with the following considerations:

- Simplicity: Although we consider that LAL tags should be as simple as possible so that humans will want to try annotating documents manually, we must offer an assisting tool for annotation in practice. The simplicity is also important to make an easy-to-use annotation tool, since if we use a feature-rich tag set, a user must check many annotation items. Therefore, the main part of LAL consists of syntactic annotation tags for specifying boundaries at many levels, and limited semantic annotation tags for specifying limited semantic information. In practice, boundary specification with limited linguistic information can cover most NLP problems, so it is sufficiently effective for NLP programs in terms of increasing accuracy.
- Assistance with NLP Tasks: The main purpose of LAL is to help NLP programs to perform their tasks much better. Therefore, in addition to tags for linguistic information, it should contain task-dependent instruction tags such as a tag indicating translation scope.

LAL tags are usually expressed by using XML namespaces. Their XML namespace prefix is **lal**. Since linguistic information annotation inherently has different annotation directions, linguistic annotation tags may overlap with other HTML and XML tags. In this case, LAL tags are expressed in the form of the processing instructions.

2.2 LAL Tags

LAL tags are classified into linguistic information tags and task-dependent instruction tags. Linguistic information tags are further classified into syntactic and semantic tags. Each type of LAL tag is described below.

2.2.1 Syntactic Information Tags

This category has tags for sentences, words, and phrases. These tags are mainly used to specify a scope for each unit.

Sentence: The sentence tag **s** is used to specify a sentence scope.

```
⟨lal:s⟩This is the first sentence.⟨/lal:s⟩
⟨lal:s⟩This is the second sentence.⟨/lal:s⟩
```

The attribute *type="hdr"* means that the sentence is a title or header.

Word: The word tag **w** is used to specify a word scope. It can have attributes for additional information such as base-form (*lex*), part-of-speech (*pos*), features (*ftrs*), and sense (*sense*) of a word. The values of these attributes are language dependent, and are not described in this paper due to the space limitation.

> ⟨lal:s⟩
> ⟨lal:w lex="this" pos="det"⟩This⟨/lal:w⟩
> ⟨lal:w lex="be" pos="verb" ftr="sg,3rd"⟩is ⟨/lal:w⟩
> ⟨lal:w lex="a" pos="det"⟩a⟨/lal:w⟩
> ⟨lal:w lex="pen" pos="noun" ftr="sg,count"⟩ pen⟨/lal:w⟩
> ⟨/lal:s⟩

The dependency (or word-to-word modification relationship) can be expressed by using the *id* and *mod* attributes of a word tag, that is, each word can have an ID value of its modifiee in a mod attribute. The ID value of a mod attribute must be an ID value of a word or a seg tag. For instance, the following example contains attributes showing that the word "with" modifies the word "saw," and which means that "she" has a telescope.

> She ⟨lal:w id="w1" lex="see" pos="v" sense="see1"⟩saw⟨/lal:w⟩ a man ⟨lal:w mod="w1"⟩with⟨/lal:w⟩ a telescope.

The *ref* attribute has the ID value of the referent of the current word. This can be used to specify a pronoun referent, for instance:

> ⟨lal:s⟩He bought a new ⟨lal:w id="w1"⟩car ⟨/lal:w⟩ yesterday.⟨/lal:s⟩
> ⟨lal:s⟩She was very surprised to learn that ⟨lal:w ref="w1"⟩it⟨/lal:w⟩ was very expensive.⟨/lal:s⟩

Phrase: The phrase tag **seg** is used to specify a phrase scope in any level. The following example specifies the scope of a noun phrase "a man ... a telescope," and this also implies that a prepositional phrase "with a telescope" modifies a noun phrase "a man."

> She saw ⟨lal:seg⟩a man with a telescope⟨/lal:seg⟩.

In addition to boundary specification, you can specify syntactic category for a phrase by using an optional attribute *cat*. The value of the cat attribute is also dependent on languages and systems. The following example specifies that a phrase "a man with a telescope" is a noun phrase.

> He saw ⟨lal:seg cat="np"⟩a man with a telescope⟨/lal:seg⟩.

The attribute *para="yes"* means that this segment also means a scope of coordination. The following example shows that a word "software" and a word "hardware" are coordinated.

> This company deals with ⟨lal:seg cat="np" para="yes"⟩software and hardware⟨/lal:seg⟩ of computer.

2.2.2 Semtantic Information Tags

LAL has the following limited semantic tags which are selected since these expressions are often used.

The **proper** tag is used to specify a proper name, and it has the *type* attribute specifying a sub-class of a proper name, such as person, place, organization, or country.

> ⟨lal:proper type="country"⟩Luxembourg ⟨/lal:proper⟩

This information is effective for translation, for instance, to select an appropriate translation word of a verb which may be changed if a subject of the verb has a human property, etc.

You can also use **acronym** and **abbr** elements defined in HTML to specify an acronym and an abbreviation terms. They are a little bit extended to have the expan attribute to specify an expanded form of abbreviation or acronym like the abbr tag of TEI[1]

> ⟨lal:acronym expan="International Business Machines"⟩IBM⟨/lal:acronym⟩

The **date** tag is used to specify a date expression, whereas, the **time** tag is used to specify a time expression. The *value* attribute is used to specify a normalized form of a date or time defined by ISO 8601 [5].

> ⟨lal:date value="2000-01-01"⟩Jan. 1, 2000 ⟨/lal:date⟩
> ⟨lal:time value="15:00"⟩3:00 PM⟨/lal:time⟩

The **num** tag is used to specify a number expression (e.g., two million and twenty-one). The *type* and *value* attributes are used to specify a normalized form of the number expression. Further, the **money** tag is used to specify money expression, in particular, to add monetary unit information.

[1] Some of LAL tags have the same name as those defined in previous efforts such as TEI, since we do not like to introduce new tag names, rather, would like to reuse existing names if the meaning is the same.

⟨lal:num type="cardinal" value="21"⟩twenty one⟨/lal:num⟩
⟨lal:money unit="usd"⟩ ⟨lal:num value="1000"⟩ one thousand ⟨/lal:num⟩ dollars ⟨/lal:money⟩

2.2.3 Task-Dependent Instruction Tags

Machine Translation: For machine translation of HTML or XML documents, we need unique algorithms to detect which segments are to be translated and which are not. In particular, XML can introduce new tags, whose semantics we generally do not know. Therefore, we need an instructional tag to inform a machine translation program whether or not a text segment is to be translated.

If an MT program encounters ⟨lal:tranStop/⟩, it passes over the subsequent text until it encounters ⟨lal:tranStart/⟩.

Text Summarization: Automatic text summarization programs have problem in handling HTML texts with the result that unimportant sentences are included in the summary texts. This problem occurs because the program extracts important sentences whose importance it calculates on the basis of the number of important keywords, the location in a text, and so on [16]. Thus, a summary program may select unimportant sentences if it does not know the main text area in a document. A typical HTML text has related information areas such as a list of related links, the name of the reporter, and a copyright notice, in the beginning and ending area, and these areas can cause a wrong summary to be generated. Therefore, we need a tag that specifies which segments should be processed in order to generate a summary of a document.

If a summary program encounters ⟨lal:smrycalcStop/⟩, it stops summary calculation until it encounters ⟨lal:smrycalcStart/⟩. Therefore, additional information parts such as a copyright notice, and a writer's signature, should not be included in this summary calculation scope.

3 LAL-aware NLP Programs

We have modified some NLP systems to be LAL-aware[2].

ESG [7, 8] is an English parsing system developed by IBM Watson Research Center, and updated to accept and generate LAL-annotated English. This LAL-aware version of ESG is used as a backend process to show users an interpretation of a system of a given English sentence in the LAL-annotation editor described in the next section.

KNP [6] is a Japanese dependency parsing system developed by Kyoto University. We have developed a post-process routine to convert KNP parsing result into LAL format. This is also used as a backend process to show the initial interpretation of a given Japanese sentence in the LAL-annotation editor.

Further, we have modified IBM's English to German, French, Spanish, and Italian translation engines [8, 9, 10] and English to Japanese translation engine [13, 14, 17] to accept LAL-annotated English HTML input.

In addition, we have developed an algorithm for accelerating CFG-parsing process by using LAL tag information[3] [19], and this algorithm is implemented in the English-to-Japanese translation engine mentioned above.

4 LAL-Annotation Editor

Since inserting tags into documents manually is not generally an easy task for end users, it is important to provide a GUI-based annotation editor. In developing such an editor, we took into consideration the following points:

- Users should not have to see any tags.
- Users should not have to see internal representations expressing linguistic information.
- Users should be able to view and modify linguistic information such as feature values, but only if they want.

With respect to the above points, we have found that most of the errors made by NLP programs result from their failure to recognize the linguistic structures of sentences. Therefore, the LAL editor shows only a structural view of a given sentence; other information is shown only if the user requests it.

The important issue here is how to represent the syntactic structure of a sentence to the user. NLP programs normally deal with a linguistic structure by means of a syntactic tree, but such a structure is not necessarily easy for end users to understand. For instance, Fig. 3 shows the dependency structure of the English sentence "IBM announced a new computer system for children with voice function." This dependency structure is not easy to understand for end users, partly because it is difficult to remind the original sentence quickly due to

[2] They support only syntactic information tags currently, and will support other tags later.

[3] This does not depend on LAL, rather it is a general algorithm applicable for CFG-parsing when any dependency information is given.

not keeping the surface word order in a given sentence in this structure[4]. Therefore, the necessary property of a linguistic structural view is for users to easily reconstruct the original surface sentence string.

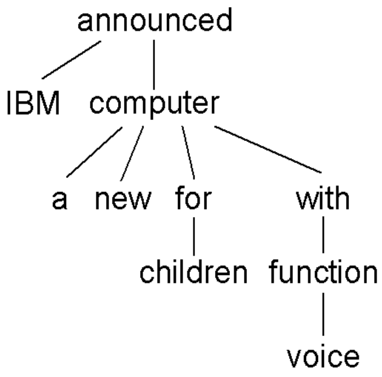


Figure 3: An example of tree structure of an English sentence

Considering this requirement, we have developed an algorithm to show linguistic dependency structure with keeping the surface word order which shows dependencies by indentations. Fig. 5 shows examples of linguistic structural view by this algorithm. In these examples, you can easily reconstruct the surface sentence string by just looking at words from top to bottom and from left to right, and easily know dependencies of words at the same time.

The next important issue is that how easily a user can understand the overall linguistic structure. If a user is, at first, presented with detailed linguistic structure in the word level, then it is difficult to grasp the important linguistic skeleton of a sentence. Therefore, another necessary property is to give users a view in which the overall sentence structure is easily recognized.

To suffice this requirement, we have introduced two presentation modes: the reduced presentation view and the expanded presentation view. In the reduced presentation view, a main verb and its modifiers are basic units for presenting dependencies, and they are located in different lines with keeping the surface order. Fig. 5 (a) shows an example of this reduced presentation view. In this view, since the obvious dependencies for native speakers (e.g. ”a” and ”computer”) are not displayed explicitly, a user can concentrate on dependencies between key units (or phrases). If a user find any dependency errors in the reduced view, he or she can enter the expanded view mode in which all words are basic units for presenting dependencies. Fig. 5 (b) and (c) shows examples of this expanded view.

```
Locate the root at an appropriate position;
Add the root to node-list;
while node-list $\neq \phi$ {
    curunit $\leftarrow$ remove-first-element(node-list);
    curline $\leftarrow$ the row of curunit;
    Add pre-modifiers of curunit to mod-list and
    sort it by the distance with curunit in the ascending order;
    while mod-list $\neq \phi$ {
        mod $\leftarrow$ remove-first-element(mod-list);
        If the forward modification is major in the
        current language, mod is the nearest premodifier, and there is no words between
        mod and curunit, then {
            Locate mod just before curunit;
        } else {
            Insert a new row just before the row of
            the curline, and make it curline;
            Locate mod in curline at the column
            after that in which the last character
            of curunit is located.
        }
    }
    curline $\leftarrow$ the row of curunit;
    Add post-modifiers of curunit to mod-list and
    sort it by the distance with curunit in the ascending order;
    while mod-list $\neq \phi$ {
        mod $\leftarrow$ remove-first-element(mod-list);
        If the backward modification is major in
        the current language, mod is the nearest
        post-modifier, and there is no words between mod and curunit, then {
            Locate mod just after curunit;
        } else {
            Insert a new row just after the row of
            the curline, and make it curline;
            Locate mod in curline at the column
            after that in which the last character
            of curunit is located.
        }
    }
}
The root unit and its direct modifiers are adjusted
to be located in the same column.
```

Figure 4: Algorithm for presenting linguistic structure

The algorithm for presenting linguistic structures we have developed is shown in Fig. 4. In this algorithm, please note that main verbs and its modifier clauses are used as presentation units (modifiees and modifiers) in the reduced view, and words are used as presentation units in the expanded view.

We have developed a GUI-based LAL-annotation editor that provides a structural views by using the above algorithm. Fig. 5 shows screen im-

[4] You must perform an inorder tree walk to reconstruct a surface sentence string.

ages of the editor. In the reduced view (as shown in (a)), an end user can easily grasp the overall structure so that "IBM" modify "announced," the phrase "a new computer" modifies (or is an direct object of) "announced," and the phrase "with voice recognition function" modifies "announced," etc. In this case, since the dependencies between "for" and "announced," and "with" and "announced" are wrong, a user changes the mode to the expanded view (as shown in (b)). In this view, a user can change dependencies by dragging a modifier to the correct modifiee using a mouse. The corrected dependency structure is shown in (c).

Fig. 6 shows the output of LAL editor for the above English sentence.

This algorithm is language-independent except for determining if forward modification or backward modification is major. Fig. 7 shows a screen image of the LAL editor for a Japanese sentence which is a translation of the above English sentence.

5 Discussion

There have been several efforts to define tags for describing language resources, such as the Text Encoding Initiative [15], OpenTag [11], Corpus Encoding Standard [1], the Expert Advisory Group on Language Engineering Standards [2], Global Document Annotation (or GDA) [3]. The main focus of these efforts other than GDA has been to share linguistic resources by expressing them in a standard tag set, and therefore they define very detailed levels of tags for expressing linguistic details. GDA has almost the same purposes but it has also defined very complex tag set. This complexity discourages people from using these tag sets when writing documents, and it becomes difficult to make an assisting tool for annotating the tags. However, LAL is not opposed to these previous efforts, but rather proposes a certain level of subset of the tags that can be used widely. In addition to this objective, as mentioned earlier, LAL's main objective is to help make NLP programs very accurate. Therefore, LAL includes task-specific annotations.

There has been some discussions about the merits of linguistic annotation tags for ordinary people. For instance, Hashida [4] stated that wide usage of such tags would greatly improve the results of NLP programs for applications such as machine translation, information retrieval, information extraction, summarization, question-answering system, example-based reasoning, and data mining, and that this would encourage ordinary people to use linguistic annotation tags. Some NLP researchers expect that since many users create HTML pages even without HTML editing tools, such users may therefore use linguistic annotation tags as well. However, it has also been observed that ordinary people write HTML pages because there is a direct advantage to them in being able to create attractive pages and an indirect advantage that the more attractive their pages, the more "hits" they will get. In contrast, linguistic annotation tags offer ordinary people only indirect advantages. Therefore, to popularize these tags, it is important to minimize the workload of adding linguistic annotation tags; that is to say, we must provide easy-to-use annotation tools. The key points in making such tools easy to use are, as mentioned earlier, minimum interaction and effective presentation. To satisfy these requirements, it is important to define a comprehensive, simple set of annotation tags.

6 Conclusion

In this paper, we have proposed an XML-compliant tag set called Linguistic Annotation Language or LAL, which helps NLP programs perform their tasks more correctly. LAL is designed to be as simple as possible so that humans can use it with minimal help from assisting tools. We have also developed a GUI-based LAL annotation editor. We hope that wide acceptance of LAL will make it possible to use more intelligent Internet tools and services.

References

[1] CES, "Corpus Encoding Standard (CES)," (http://www.cs.vassar.edu/CES/)

[2] EAGLES, "Expert Advisory Group on Language Engineering Standards," (http://www.ilc.pi.cnr.it/EAGLES/home.html)

[3] GDA, "Global Document Annotation," (http://www.etl.go.jp/etl/nl/gda/)

[4] Koichi Hashida, Katashi Nagao, et. al, "Progress and Prospect of Global Document Annotation," (in Japanese) Proc. of 4th Annual Meeting of the Association of Natural Language Processing, pp. 618–621, 1998

[5] "Data elements and interchange formats – Information interchange – Representation of dates and times," ISO 8601:1988.

[6] Kurohasi, S., and Nagao, M., "A Syntactic Analysis Method of Long Japanese Sentences based on the Detection of Conjunctive Structures," Computational Linguistics, Vol. 20, No. 4, 1994.

[7] McCord, C. M., "Slot Grammars," Computational Linguistics, Vol. 6, pp. 31–43, 1980.

[8] McCord, C. M., "Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars," in (ed) R. Studer, Natural Language and Logic: International Scientific Symposium, Lecture Notes in Computer Science, pp. 118–145, Springer Verlag, 1990.

```
⟨?xml version=”1.0” encoding=”US-ASCII”?⟩
⟨lal⟩
⟨lal:s id=”id1”⟩
⟨lal:w id=”id1-1” mod=”id1-2” pos=”noun” lex=”IBM” ftrs=”sg,propn”⟩IBM ⟨/lal:w⟩
⟨lal:w id=”id1-2” pos=”verb” lex=”announce”⟩announced ⟨/lal:w⟩
⟨lal:w id=”id1-3” mod=”id1-5” pos=”det” lex=”a” ftrs=”sg”⟩a ⟨/lal:w⟩
⟨lal:w id=”id1-4” mod=”id1-5” pos=”adj” lex=”new”⟩new ⟨/lal:w⟩
⟨lal:w id=”id1-5” mod=”id1-2” pos=”noun” lex=”computer” ftrs=”sg,cn”⟩computer ⟨/lal:w⟩
⟨lal:w id=”id1-6” mod=”id1-5” pos=”prep” lex=”for”⟩for ⟨/lal:w⟩
⟨lal:w id=”id1-7” mod=”id1-6” pos=”noun” lex=”child”⟩children ⟨/lal:w⟩
⟨lal:w id=”id1-8” mod=”id1-5” pos=”prep” lex=”with”⟩with⟨/lal:w⟩
⟨lal:w id=”id1-9” mod=”id1-10” pos=”noun” lex=”voice”⟩voice ⟨/lal:w⟩
⟨lal:w id=”id1-10” mod=”id1-11” pos=”noun” lex=”recognition”⟩recognition⟨/lal:w⟩
⟨lal:w id=”id1-11” mod=”id1-8” pos=”noun” lex=”function”⟩function⟨/lal:w⟩
.
</lal:s⟩
⟨/lal⟩
```

Figure 6: Example of LAL Annotation Output

[9] McCord, C. M., “Heuristics for Broad-Coverage Natural Language Parsing,” Proc. of the ARPA Human Language Technology Workshop, 1993.

[10] McCord, C. M., and Bernth, A., “The LMT Transformational System,” Proc. of Proceedings of AMTA-98, pp. 344–355, 1998.

[11] OpenTag, “A Standard Extraction/Abstraction Text Format for Translation and NLP Tools,” (http://www.opentag.org/)

[12] SGML, “ISO/IEC 8879-1986 (E). Information processing – Text and Office Systems – Standard Generalized Markup Language (SGML). First Edition – 1986-10-15.International Organization for Standardization,” 1986.

[13] Takeda, K., “Pattern-Based Context-Free Grammars for Machine Translation,” Proc. of 34th ACL, pp. 144–151, June 1996.

[14] Takeda, K., “Pattern-Based Machine Translation,” Proc. of 16th COLING, Vol. 2, pp. 1155–1158, August 1996.

[15] TEI, “Text Encoding Initiative (TEI),” (http://www.uic.edu:80/orgs/tei/)

[16] Watanabe, H., “A Method for Abstracting Newspaper Articles by Using Surface Clues,” Proc. of 16th International Conference of Computational Linguistics, pp. 974–979, Aug. 4-9, 1996.

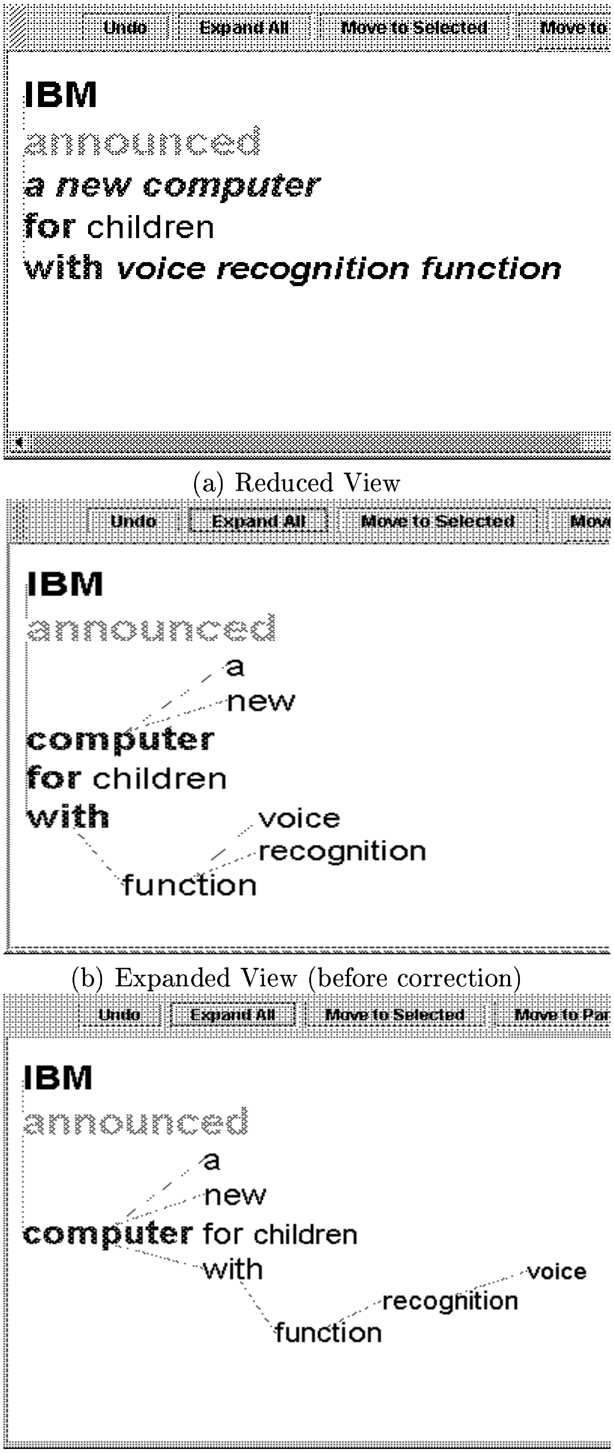
[17] Watanabe, H., and Takeda, K., “A Pattern-based Machine Translation System Extended by Example-based Processing,” Proc. of the 36th ACL & 17th COLING, Vol. 2, pp. 1369-1373, 1998.

[18] Watanabe, H., “Linguistic Annotation Language – The Markup Language for Assisting NLP programs –,” IBM Research Report RT0334, 1999.

[19] Watanabe, H., “A Method for Accelerating CFG-Parsing by Using Dependency Information,” Proc. of 18th COLING, 2000.

[20] XML, “Extensible Markup Language (XML),” (http://www.w3.org/TR/PR-xml-971208), World Wide Web Consortium, Dec. 8, 1997.

[21] XMLNS, “Namespaces in XML,” (http://www.w3.org/TR/1998/WD-xml-names-19980327), World Wide Web Consortium, March 27, 1998.



(a) Reduced View

(b) Expanded View (before correction)

(c) Expanded View (after correction)

Figure 5: Screen Image of LAL Editor for English sentence

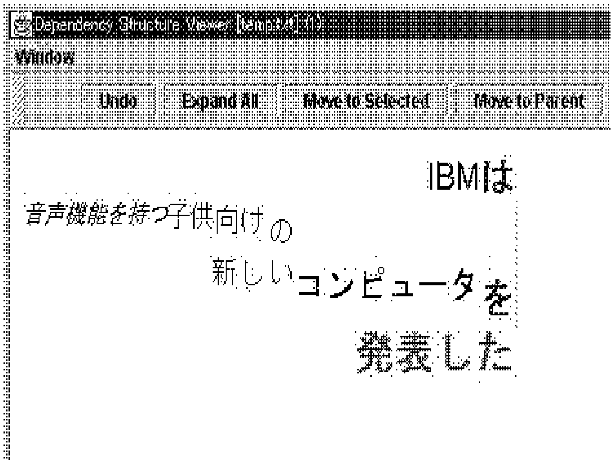


Figure 7: Screen Image of LAL Editor for Japanese sentence

Building an Annotated Corpus in the Molecular-Biology Domain

Yuka Tateisi, Tomoko Ohta, Nigel Collier, Chikashi Nobata, Jun-ichi Tsujii
Department of Information Science
Graduate School of Science
University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

Abstract

Corpus annotation is now a key topic for all areas of natural language processing (NLP) and information extraction (IE) which employ supervised learning. With the explosion of results in molecular-biology there is an increased need for IE to extract knowledge to support database building and to search intelligently for information in online journal collections. To support this we are building a corpus of annotated abstracts taken from National Library of Medicine's MEDLINE database. In this paper we report on this new corpus, its ontological basis, and our experience in designing the annotation scheme. Experimental results are shown for inter-annotator agreement and comments are made on methodological considerations.

1 Introduction

In the field of molecular biology there have recently been rapid advances that have motivated researchers to construct very large databases in order to share knowledge about biological substances and their reactions. A large part of this knowledge is only available in unformalized research papers and information extraction (IE) from such sources is becoming crucial to help support timely database updating and to help researchers avoid problems associated with information overload.

For this purpose, various NLP techniques have been applied to extract substance names and other terms (Ohta et al., 1997; Fukuda et al., 1998; Proux et al., 1998; Nobata et al., 1999) as well as information concerning the nature and interaction of proteins and genes (Sekimizu et al., 1998; Blaschke et al., 1999; Hamphrays et al., 2000; Thomas et al., 2000; Rindflesch et al., 2000). The nomenclatures of genes and associated proteins for model organisms such as *S. Cerevisiae* (yeast) and *D. Melanogaster* (fruit fly) are established so that good dictionaries for those names have been constructed. However nomenclatures for humans are not yet available as the whole picture of the human genome has yet to be revealed, this results in arbitrary names being used by researchers who identified the structure of proteins and genes, so dictionary-based approaches might not be as effective as in the case of model organisms. Thus many of the previous researchers either limit their scope to extracting information on substances like enzymes which have established naming conventions (Hamphrays et al., 2000) or extracting information on 'substance' giving up the distinction between the class of substance like protein and DNA (Fukuda et al., 1998; Proux et al., 1998; Sekimizu et al., 1998; Thomas et al., 2000).

Term identification and classification methods based on statistical learning seem to be more generalizable to new knowledge types and representations than the methods based on dictionaries and hand-constructed heuristic rules. We think that a corpus-based, machine-learning approach is quite promising, and to support this we are building a corpus of annotated abstracts taken from National Library of Medicine (NLM)'s MEDLINE database.

Corpus annotation is now a key topic for all areas of natural language processing and linguistically annotated corpus such as treebanks are now established. In information extraction task, annotated corpora have been made mainly for the judgment set of information extraction competitions such as MUC (Chinchor, 1998). We think that technical terms of a scientific domain share common characteristics with the "Named Entities" and the tasks we attempt involve recognition and classification of the names of substances and their locations, just as named entity recognition task in MUC conferences. We therefore try to model our annotation task after the definition of "EnameX" (Chincor, 1998a) of MUC conferences. Unlike in MUC conferences, we don't make a precise definition of how the recognized names are used in further information extraction task such as event identification, because we want the recognition technology to be independent of the further task. Our work is also compared to word-sense annotation (e.g.,(Bruce and Wiebe, 1998)) where instances of words that have multiple senses are labelled for the sense it denotes according to a certain dictionary or thesaurus.

We first built a conceptual model (ontology) of substances and sources (substance location), and designed a tag set based on the ontology which conforms to SGML/XML format. Using the tag set, we annotated the entities such names that appears in the abstracts of research papers taken from the MEDLINE database. In this paper we report on this new corpus, its ontological basis, and our experience in designing the annotation scheme. Experimental results are shown for inter-annotator agreement and comments are made on methodological considerations.

2 Design of The Tag Set

2.1 Underlying Ontology

The task of annotation can be regarded as identifying and classifying the names that appears in the texts according to a pre-defined classification. For a reliable classification, the classification must be well-defined and easy to understand by the domain experts who annotate the texts. To fulfill this requirement, we create a concrete data model (ontology) of the biological domain on which the tag sets are based.

Ontologies have been developed in the biomedical sciences for several applications. Such ontologies include conceptual hierarchies for databases covering diseases and drug names. Construction of a more general ontology e.g. (Baker et al., 1999) is being attempted by several groups interested in interconnecting databases under a uniform view.

We start from a taxonomy illustrated in Figure 1[1]. In this taxonomy, we classify substances according to their *chemical* characteristics rather than their biological role. This is unlike other existing ontologies in the biology field (Baker et al., 1999; Schulze-Kremer, 1998), which mix the classification by biological role and by chemical structure. The reason that we have adopted this approach is that we consider mixing two criteria prevents the mutually exclusive classification and thus makes the annotated task more complicated by introducing nested tag structures and context dependent semantic tags. In our initial annotation work we therefore chose to simplify the classification by concentrating on the chemical structure.

Chemical classification of substances is quite independent of the biological context in which it appears, and is therefore more stably defined. For example, the chemical characteristics of a protein can be easily defined, but its biological role may vary depending on the biological context, e.g., it may work as an enzyme for one species but a poison for others. Therefore, in our model we do not classify substance as enzymes, transcription factors, genes, etc. but as proteins, DNAs, RNAs, etc. They are further classified into families, complexes, individual molecules, subunits, domains, and regions, because these super- and sub- structures often have separate names. This classification is non-controversial among biologists and can be easily expanded into other ontologies.

Sources are biological locations where substances are found and their reactions take place, such as *human* (an organism), *liver* (a tissue), *leukocyte* (a cell), *membrane* (a sub-location of a cell) or *HeLa* (a cultured cell line). Organisms are further classified into multi-cell organisms, mono-cell organisms other than viruses, and viruses. Organism, tissue, cell, sub-locations are interrelated with *part-of* relation but that relation is not shown in Figure 1. Based on this domain model, we annotate the names of proteins, DNAs, RNAs, and sources using the tags shown in Table 1.

An example of an annotated text is shown in Figure 2: the `UI` number is a unique identifier of the abstract in MEDLINE assigned by

[1] In Figure 1 the concepts represented in **bold** are reflected in the tag set and the concepts represented in *italic* are reflected in the attributes.

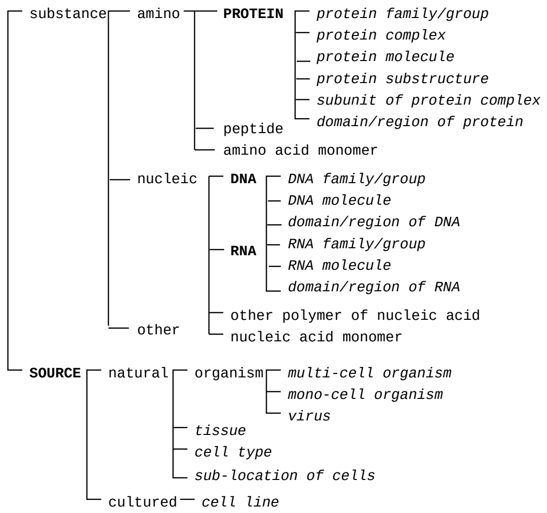


Figure 1: The taxonomy used as a domain model of our tagging scheme

Table 1: Tags and their target objects

tag	object
<PROTEIN>	the names of proteins, including protein groups, families, molecules, complexes, and substructures
<DNA>	the names of DNAs, including DNA molecules, DNA groups, DNA regions, and genes
<RNA>	the names of RNAs, including DNA molecules, RNA groups, RNA regions, and genes
<SOURCE>	the sources of substances, i.e., the names of organisms, tissues, cells, sub-locations of cells, and cell lines

the National Library of Medicine, `TI` is the title, and `AB` is the abstract text. The `unsure` attribute shown in the text is optional. This is used when annotators are unsure about whether a name should be tagged or whether the boundary of the tagged name is correct, and when the annotator was sure about the instance of the markup, `unsure` attribute can be omitted (or can be assigned the value `ok`).

3 Tagging Task

```
UI - 91012785
TI - <PROTEIN unsure=ok>Lymphotoxin</PROTEIN>
activation by <SOURCE subtype=cl unsure=ok>human
T-cell leukemia virus type I-infected cell
lines</SOURCE>:  role for <PROTEIN unsure=ok>NF-kappa
B</PROTEIN>.  AB - <SOURCE subtype=cl
unsure=ok>Human T-cell leukemia virus type
I (HTLV-I)-infected T-cell lines</SOURCE>
constitutively produce high levels of biologically
active <PROTEIN unsure=ok>lymphotoxin</PROTEIN>
(<PROTEIN unsure=ok>LT</PROTEIN>; <PROTEIN
unsure=ok>tumor necrosis factor-beta</PROTEIN>)
protein and <RNA unsure=ok>LT mRNA</RNA>.
To understand the regulation of <PROTEIN
unsure=ok>LT</PROTEIN> transcription by <SOURCE
subtype=vi unsure=ok>HTLV-I</SOURCE>, we analyzed
the ability of a series of deletions of the
<DNA unsure=ok>LT promoter</DNA> to drive the
<DNA unsure=ok>chloramphenicol acetyltransferase
(CAT) reporter gene</DNA> in <SOURCE subtype=cl
unsure=ok>HTLV-I-positive MT-2 cells</SOURCE>.  The
smallest <DNA unsure=ok>LT promoter fragment</DNA>
(-140 to +77) that was able to drive CAT activity
contained a site that was similar to the <DNA
unsure=ok>immunoglobulin kappa-chain NF-kappa
B-binding site</DNA>.
```

Figure 2: Example of Annotated Text

Before beginning the tagging process we made a preliminary experiment by tagging 100 abstracts. The abstracts were 116 words long on average. One of the authors, who has a doctorate in molecular biology, manually tagged the abstracts. The process took about 40 hours. 2125 proteins, 358 DNAs, 30 RNAs, and 801 SOURCEs are tagged.

Ten abstracts out of the 100 were randomly chosen and three other volunteers, two medical science researchers and one biology researcher, were asked to annotate them with our tagging scheme. We gave a brief explanation on the tagging task and scheme to each annotator. The annotators were asked to annotate the text independently in one weeks' time.

After the annotation was done, we sent a questionnaire to annotators to ask for their comments on the tagging task and the guide. From the feedback of the questionnaire, we learned that the annotators felt the task to be relatively easy, but there are several cases where the they were unsure about which tags to be assigned where. The cases include:

- where two or more names are conjoined with *and* or *or*, e.g., `IRF-1 mRNA and protein`
- the ambiguity in some papers concerning

Table 2: The percentage of inter-annotator agreement on 10 abstracts

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Mean
A0-A1	100.00	69.05	38.18	82.76	69.81	83.87	74.07	83.33	88.31	91.67	77.29
A0-A2	100.00	60.98	66.13	67.65	80.49	72.31	72.73	90.11	84.21	71.43	76.78
A0-A3	95.24	59.09	57.63	96.55	86.05	83.82	69.64	79.55	85.71	84.91	78.44
A1-A2	100.00	83.78	41.18	60.61	62.00	67.21	83.02	77.65	80.82	78.72	72.55
A1-A3	95.24	52.50	66.67	78.57	61.54	76.56	77.78	70.73	79.73	76.47	72.76
A2-A3	95.24	51.28	47.27	63.64	85.00	76.12	85.45	82.02	83.56	65.38	73.85
Mean	97.62	62.78	52.84	74.96	74.15	76.65	77.16	80.57	83.72	78.10	75.85

whether names denote DNAs, RNAs or proteins,

The annotators also said that the concrete example of tagged texts are more useful than descriptions and more examples should be included in the manual.

Two-way agreement rate is scored according to the scheme used in MUC conferences(Chincor, 1998b). This scoring scheme uses the F-measure derived from recall and precision. Recall R and precision P are given by:

$$R = |X \cap Y|/|X| \tag{1}$$

and

$$P = |X \cap Y|/|Y| \tag{2}$$

where X is the set of 'correct' objects and Y is the set of 'retrieved' objects. The F-measure is the harmonic mean of R and P given by

$$F = 1/(1/P + 1/R) = 2 \times |X \cap Y|/(|X| + |Y|) \tag{3}$$

and this F can be used to measure the agreement of two sets of objects neither of which are considered 'correct' (note that F is symmetric with regards to X and Y).

The F-measures multiplied by 100 to show the percentage of the agreement between annotators for the 10 abstracts are shown in Table 2. In Table 2, `T1, ..., T10` denotes the abstracts and `A0, ..., A3` denotes annotators. The table shows that the agreement rate, comparable to man-machine agreement of systems participated in MUC, is not good for inter-annotator agreement rate. The disagreement indicate that there are several problems in the definition of the target and the description in the manual, some of which seem to be specific to this domain[2].

We investigate into the case of disagreement by aligning the tagged text and examining the disagreed parts by hand. We found that the disagreement could be classified into several patterns enlisted below. The numbers in the parentheses in the items are the number of the occurrence of the disagreement in total 10 texts. See Table 3 for examples[3].

Division (27): The cases where a same part of a text is tagged as one by some annotators but divided into two (or more) parts by others. They were further classified into the following cases.

D-1 (13) parenthesized abbreviations, full forms, and synonyms
D-2 (3) appositive phrases
D-3 (6) names of a substance which includes `SOURCE` names
D-4 (2) names of a complex
D-5 (3) conjoined names

Part (60): The cases where a part of phrases is included between `<TAG>` and `</TAG>` by some annotators but not by others. They were further classified into the following cases.

P-1 (30) the cases where the substances designated by the tagged part are changed by whether the words following a name are tagged together or not: in 10 cases, different tags are used by the annotators; in

[2] Though it may not be directly compared, inter-annotator agreement for the judgment set of IREX conference on Japanese information extraction(Sekine, 1999) is reported to be around 97% in F-measure.

[3] Attributes are omitted in the examples.

Table 3: Examples of disagreement

Cases	Examples
D-1	<SOURCE>Mycobacterium avium complex (MAC)</SOURCE> <SOURCE>Mycobacterium avium complex</SOURCE> (<SOURCE>MAC</SOURCE>)
D-2	<SOURCE>U937, a human monocytoid cell line</SOURCE> <SOURCE>U937</SOURCE>, <SOURCE>a human monocytoid cell line</SOURCE>
D-3	<PROTEIN>Human erythroid 5-aminolevulinate synthase</PROTEIN> <SOURCE>Human erythroid</SOURCE> <PROTEIN>5-aminolevulinate synthase</PROTEIN>
D-4	<PROTEIN>p50-p65</PROTEIN> <PROTEIN>p50</PROTEIN>-<PROTEIN>p65</PROTEIN>
D-5	<RNA>ferritin or transferritin receptor mRNAs</RNA> <PROTEIN>ferritin</PROTEIN> or <RNA>transferritin receptor mRNAs</RNA>
P-1 (different tags)	<DNA>AP-2 consensus binding sequences</DNA> <PROTEIN>AP-2</PROTEIN> consensus binding sequences
P-1 (same tags)	<PROTEIN>IRF-2 repressor</PROTEIN> <PROTEIN>IRF-2</PROTEIN> repressor
P-2	<PROTEIN>Stat91 protein </PROTEIN> <PROTEIN>Stat91</PROTEIN> protein
P-3	<RNA>housekeeping ALAS mRNA</RNA> housekeeping <RNA>ALAS mRNA</RNA>
P-4	<PROTEIN>transcription factor AP-2</PROTEIN> transcription factor <PROTEIN>AP-2</PROTEIN>
P-5	<DNA>the terminal protein 1 gene promoter</DNA> the <DNA>terminal protein 1 gene promoter</DNA>
Class	<RNA>TAR</RNA> <DNA>TAR</DNA>
Missing	<DNA>21 bp repeats</DNA> 21 bp repeats

the other 20 cases, the same tags are used by the annotators.

P-2 (18) the cases where the substances designated by the tagged part are not affected by whether the words following a name are tagged together or not

P-3 (6) the preceding attributive phrase that narrows the meaning of the phrase

P-4 (5) the preceding appositive phrase

P-5 (1) determiners

Class (19): The same part of text is tagged with different tags

Missing (25): A part of text is tagged by some annotators but not by others

The result shows that most of disagreement involves recognizing the names, i.e., identifying the range of words in sentence that are part of the names. On the other hand, there are relatively few cases where classification of the names alone is the problem.

The disagreement involving abbreviation and synonym (case D-1) will be simply solved by explicitly giving an instruction as to whether a full form and its abbreviation (or a name and its synonym) should be separated or not. The case of appositives (cases D-2 and P-5) and determiners are also easy to solve by giving explicit instruction, though the distinction between appositives or determiners and other attributive phrases (case P-4) must be carefully stated in the instruction. The cases involving words that follow a name that do not affect the substance the name designates (P-2) should be handled similarly with a careful description of such cases in the instruction.

The cases that involve the source names (case D-3) and the following words that modify the meaning of the phrase (P-1) are more difficult, because the names with or without the modifying phrases are recognized by the annotators. One solution would be to allow nesting tags, but this might complicate the tagging scheme and be the cause of another type of error. Simple heuristics of 'taking the longest phrase' might work here, but in the case of preceding modifiers (P-3) the heuristic is not desirable, because most of the preceding modifiers are just description of a characteristics of a substance.

The names tagged by some annotators but not by others (case M) were mostly the terms that describes the parts of a gene as in the example above, or the terms that denotes a family or a class of substances. Such parts or families are considered to be the 'substance' by some annotators but not by others. Incorporating the distinction between families, individual substances, and parts of the substance would help to make the classification of names clearer and result in more consistent annotation.

One of the difficulties of this task compared to MUC named entity extraction is that our targets are inherently unique names of classes, whereas the targets of MUC named entity extraction are names of unique entities. When we refer to a specific protein or DNA, we don't refer to a specific molecule, but rather a class of molecules that have the same characteristics. As the name of a class, when a researcher finds a new substance, the substance is often named after the combination of its function, location, etc. For example, "B-cell specific transcription factor" is a name of a protein (there is an entry in the SwissProt database). This results in the difficulty of distinguishing the names of substances from general description of the substance. In cases such as "Human erythroid 5-aminolevulinate synthase", some researchers recognize it as a name but some only recognize "5-aminolevulinate synthase" as a name and "Human erythroid" as just a description and separate the part as different entity. Also the prenominal modifiers are recognized or not recognized as a part of the name depending on whether the names with or without the modifying phrases are recognized by the annotators.

The classification error, though relatively few, also might be from the nature of this domain. Most of the inconsistency are suspected to be from conventional use of the protein names to denote the genes that transcribe the protein. For example, `NF-kappa B gene` is a name of a gene that transcribe the protein NF-kappa B, and the authors often omit the word `gene` where they think it is clear from the context that the particular occurrence of `NF-kappa B` denotes the DNA. This require the annotators good background knowledge and careful reading, and sometimes the cause of annotation errors. Even the participated annotators, who are qualified specialist of the domain, are sometimes unsure about the target, according to the questionnaire. This might be resolved if the full paper could be referenced in the process of annotation.

4 Conclusion and Future Work

We are in the process of developing a high-quality tagging scheme for semantic annotation of substances and their sources which play an important role in molecular-biology events. We have shown the results of initial inter-annotator agreement tests using the current scheme. After the initial experiment, we revised the tagging manual to give more precise definitions and more examples, and also added attributes to denote the distinction of whether the protein (DNA, RNA) is a molecule, complex, substructure, region, etc. We tagged 500 abstracts according to the revised manual and tagging-scheme, which are in the process of cross-checking and cleaning up the errors. When they are done we plan to make the corpus available to the public along with the tagging manual.

Establishing the training process of annotators, including communication between annotators to get agreement on tagging strategies, which is reported to improve the agreement rate (Dan Melamed, 1998; Wiebe et al., 1999) should also be necessary to help them make consist annotation.

One of the concerns that we have is that our target task is more difficult than the traditional named entity recognition task, because of the naming convention (or the lack of it) of the molecular-biology domain and because the task requires very precise knowledge of the specialist. To solve this problem, tagging tools that incorporates the reference function to the external sources such as substance databases, on-line glossaries, and full-text of the paper should also be of great help.

The preliminary corpus, though it may be 'noisy', can be useful as a training set for recognition program of biological names and terms. The preliminary corpus can also be used to gain the knowledge of how the tagged names are related to each other and other names, in order to give feedback to the annotators and enhance the domain model and enables us to annotate more rich information such as biological roles.

References

P. G. Baker, C. A. Goble, S. Bechhofer, N. W. Paton, R. Stevens, and A. Brass. 1999. An ontology for bioinformatics applications. 15:510–520.

C. Blaschke, M. A. Andrade, C. Ouzounis, and A. Valencia. 1999. Automatic extraction of biological information from scientific text: protein-protein interactions. In *Proc. 7th International conference on Intelligent Systems for Molecular Biology*, pages 60–67.

R. Bruce and J. Wiebe. 1998. Word sense distinguishability and inter-coder agreement. In *Proc. 3rd Conference on Empirical Methods in Natural Language Processing*, pages 53–60.

N. Chinchor. 1998. Overview of MUC-7. In *Proceedings of 7th Message Understanding Conference*. available at http://www.muc.saic.com/proceedings.

N. Chincor. 1998a. MUC-7 named entity task definition version 3.5. In *Proceedings of 7th Message Understanding Conference*. available at http://www.muc.saic.com/proceedings.

N. Chincor. 1998b. MUC-7 test scores introduction. In *Proceedings of 7th Message Understanding Conference*. available at http://www.muc.saic.com/proceedings.

I Dan Melamed. 1998. Manual annotation of translation equivalence:the blinker project. Technical Report IRCS-98-07, IRCS, University of Pennsylvania. available at ftp://ftp.cis.upenn.edu/pub/ircs/tr/98-07/.

K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. 1998. Towards information extraction: Identifying protein names from biological papers. In *Proc. 3rd Pacific Symoisium of Biocomputing*, pages 707–718.

K. Hamphrays, G. Demetriou, and R. Gaizauskas. 2000. Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In *Proc. 5th Pacific Symoisium of Biocomputing*, pages 72–80.

C. Nobata, N. Collier, and J. Tsujii. 1999. Automatic term identification and classification in biology texts. In *Proc 5th Natural Language Processing Pacific Rim Symoisium*, pages 369–374.

Y. Ohta, Y. Yamamoto, T. Okazaki, and T. Takagi. 1997. Automatic construction of knowledge base from biological papers. In *Proc. 5th International Conference on Intelligent Systems for Molocular Biology*, pages 218–225.

D. Proux, F. Rechenmann, L. Julliard, V. Pillet, and B. Jacq. 1998. Detecting gene symbols and names in biological texts: A first step toward pertinent information extraction. In *Genome Informatics*, pages 72–80. Universal Academy Press.

T. C. Rindflesch, L. Tanabe, J. N. Weinstein, and L. Hunter. 2000. Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Proc. 5th Pacific Symposium on Biocomputing*, pages 514–525.

S. Schulze-Kremer. 1998. Ontologies for molecular biology. In *Proc. 3rd Pacific Symposium on Biocomputing*, pages 695–706.

T. Sekimizu, H. S. Park, and J. Tsujii. 1998. Identifying the interaction between genes and gene products based on frequently seen verbs in MEDLINE abstracts. In *Genome Informatics*, pages 62–71. Universal Academy Press.

S. Sekine. 1999. Analysis of the answer of named entity extraction. In *Proceedings of the IREX workshop*, pages 129–132. in Japanese.

J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll. 2000. Automatic extraction of protein interactions from scientific abstracts. In *Proc. 5th Pacific Symposium on Biocomputing*, pages 538–549.

J. Wiebe, R. Bruce, and T. O'Hara. 1999. Development and use of a gold standard data set for subjectivity classifications. In *Proceedings of the 37th Meeting of ACL*, pages 246–253.

SECTION 2

Semantic Annotation of Discourse Structure

Semantic Annotation for Generation: Issues in annotating a corpus to develop and evaluate discourse entity realization algorithms

Massimo Poesio
University of Edinburgh
HCRC and Informatics
{Massimo.Poesio}@ed.ac.uk

Abstract

We are annotating a corpus with information relevant to discourse entity realization, and especially the information needed to decide which type of NP to use. The corpus is being used to study correlations between NP type and certain semantic or discourse features, to evaluate hand-coded algorithms, and to train statistical models. We report on the development of our annotation scheme, the problems we have encountered, and the results obtained so far.

1 MOTIVATIONS

The goal of the GNOME project is to develop NP generation algorithms that can be used by real systems, with different architectures, and operating in realistic domains. As part of the project, we have been annotating a corpus with the syntactic, semantic and discourse information that is needed for different subtasks of NP realization, including the task of deciding on the most appropriate NP type to be used to realize a certain discourse entity (proper name, definite description, pronoun, etc.), and the task of organizing the additional information to be expressed with that discourse entity. We are using the annotated corpus to extract information useful to the development of hand-coded algorithms for the subtasks of NP realization we are focusing on, to develop statistical models of these subtasks, and to evaluate both types of algorithms. Conversely, we have been using the results of this evaluation to verify the completeness of our annotation scheme and to identify modifications. The annotation scheme used in our first corpus annotation exercise was discussed in (Poesio et al., 1999b); in this paper we present the modified annotation scheme that we developed as a result of that preliminary work, and discuss the problems we encountered when trying to annotate semantic and discourse information.

2 APPLICATIONS AND DATA

The systems we are working with are the ILEX system developed at HCRC, University of Edinburgh (Oberlander et al., 1998),[1] and the ICONOCLAST system (Scott et al., 1998), developed at ITRI, University of Brighton. The ILEX system generates Web pages describing museum objects on the basis of the perceived status of its user's knowledge and of the objects she previously looked at; ICONOCLAST supports the creation of pharmaceutical leaflets by means of the WYSIWYM technique in which text generation and user input are interleaved.

The corpus we have collected for GNOME includes texts from both the domains we are studying. It contains texts in the museum domain, extending the corpus collected by the SOLE project (Hitzeman et al., 1998); and texts from the corpus of patient information leaflets collected for the ICONOCLAST project. The initial GNOME corpus (Poesio et al., 1999b) consisted of two subsets of about 1,500 NPs each; since then, the corpus has been extended and currently includes about 3,000 NPs in each domain. We are also adding texts from a third domain, tutorial dialogues.

3 DEVELOPING A SCHEME FOR NP REALIZATION

The traditional approach to surface realization in NLG (as exemplified, say, by NIGEL / KPML (Henschel et al., 1999)) assumes (systemic functional) grammars that make decisions on the basis of the answer to queries asked to the knowledge base and discourse model. Typical examples of such queries are:

- whether a given discourse entity is IDENTIFIABLE;

[1] The latest version of the system can be found at http://www.cstr.ed.ac.uk/cgi-bin/ilex.cgi.

- whether the object denoted is GENERIC or not;
- whether that entity is IN FOCUS, or more generally what is its ACCESSIBILITY (Gundel et al., 1993)
- what is the ONTOLOGICAL STATUS of the object, i.e., its position in a taxonomy.

These systems have typically been used only by their developers, or by researchers working in close collaboration with them. In order to make them more generally usable, three questions have to be addressed. The first question is whether anybody other than the developers of these grammars can understand queries such as those just listed enough to implement them in their systems. The second is whether real systems have enough information to answer these queries, or whether instead approximations have to be implemented. The final question is how well the implementation is going to perform, especially if only approximations are implemented.

In GNOME we have been studying these questions by means of corpus annotation studies. We have been trying to identify which of the queries used by systems such as KPML for NP realization can be generally understood by asking subjects to annotate the NPs in our corpus with the information needed to answer these queries, and we have then used the resulting annotation to train statistical models to evaluate the completeness of a given set of features. We use to measure agreement the K statistic discussed by Carletta (1996). A value of K between .8 and 1 indicates good agreement; a value between .6 and .8 indicates some agreement.

4 SEMANTIC AND DISCOURSE FEATURES THAT MAY AFFECT NP TYPE DETERMINATION

Even if in this first phase we focused on realizing discourse entities only, we still need to know for each NP in the corpus its semantic type. Noun phrases appear in a text as the realization of at least three different types of logical form constituents:

- **terms**, which include referring expressions, as in *Jessie M. King* or *the hour pieces here* , but also non-referring terms such as *jewelry* or *different types of creative work*. Terms are called DISCOURSE ENTITIES in Discourse Representation Theory.
- **quantifiers**, as in *quite a lot of different types of creative work* or *nearly every day*
- **nominal predicates**, such as *an illustrator* in *She was an illustrator*.

Noun phrases can be **coordinated**, as in *The patches also contain oestradiol and norethisterone acetate* or *the inventory gives neither the name of the maker nor its original location*; we finesse the many issues raised by coordination by assuming a fourth type of logical form objects, **coordinations**.

Two features generally acknowledged to play an important role in determining the type of the NP to be used to realize a discourse entity are COUNTABILITY and GENERICITY. These features are especially important when bare-NPs are going to be used. One of the conditions under which (singular) bare NPs are used is when the object denoted is mass (cfr. **a gold/a jewel* vs. *gold/*jewel*); the other is when the NP is used to express a generic reference, as in *The cabinets de curiosites contained natural specimens such as shells and fossils*.

Much work on NP generation has been devoted to studying the discourse factors that determine whether a given discourse entity should be realized by a definite or an indefinite NP (Prince, 1992; Loebner, 1987; Gundel et al., 1993). Among the discourse properties of a discourse entity claimed to affect its form are

- Whether it is discourse new or old (Prince, 1992): e.g., a new jewel would be introduced by means of the indefinite *a jewel*, whereas for an already mentioned one the definite description *the jewel* would be used. This simple notion of familiarity was refined by Prince herself as well by Gundel *et al.* (Gundel et al., 1993).
- Whether it's hearer-new or hearer-old (Prince, 1992).
- Whether it is referring to an object in the visual situation or not: if so, a demonstrative NP may be used, as in *this jewel*.
- Whether it's currently highly salient or not, which may prompt the use of a pronoun. Properties that have been claimed to affect the salience of a discourse entity include: whether it's the current CENTER (CB) or not (Grosz et al., 1995), or more generally whether that entity is the TOPIC of the current discourse (Reinhart, 1981; Garrod and Sanford, 1983); its grammatical function; whether it's animated or

not; its role; its proximity. (For a discussion of the effect of these and other factors on salience see (Poesio and Stevenson, To appear)).

According to Loebner (Loebner, 1987), the distinguishing property of definites is not familiarity (a discourse notion), but whether or not the predicate denoted by the head noun is functional or, more generally, UNIQUE. This seems to be the closest formal specification of the notion of 'identifiability' used in KPML.

5 THE ANNOTATION SCHEME

Our first scheme, and the results we obtained with it, are discussed in (Poesio et al., 1999b). We are currently in the process of reannotating the corpus from scratch according to a new annotation scheme developed to address the limitations of the scheme discussed there (reliability and/or incompleteness of information). The new scheme also includes information to study another aspect of NP realization, NP modification; this aspect of the new annotation won't be discussed here. For reasons of space, only a brief discussion is possible - in particular, we won't be able to discuss in detail the instructions given to annotators; the complete instructions are available at `http://www.hcrc.ed.ac.uk/˜gnome/anno_manual.html`.

Markup Language

Our annotation scheme is XML-based. The basis for our annotation are a rather minimal set of layout tags, identifying the main divisions of texts, their titles, figures, paragraphs, and lists. Also, as a result of the reliability studies discussed below and of our first annotation effort, we decided to also mark up units of text that may correspond to rhetorical units in our second annotation, using the tag ⟨unit⟩.

An important feature of the scheme is that the information about NPs is split among two XML elements, as in the MATE scheme for coreference (Poesio et al., 1999a). Each NP in the text is tagged with an ⟨ne⟩ tag, as follows:

(1)
```
<ne ID="ne07" ...  >
Scottish-born, Canadian based jew-
eller,
Alison Bailey-Smith</ne>
...
<ne ID="ne08"> <ne ID="ne09">Her</ne>
materials</ne>
```

the instructions for identifying the ⟨ne⟩ markables are derived from those proposed in the MATE project scheme for annotating anaphoric relations (Poesio et al., 1999a), which in turn were derived from those proposed by Passonneau (Passonneau, 1997) and in MUC-7 (Chinchor and Sundheim, 1995).

Anaphoric relations are annotated by means of a separate ⟨ante⟩ element specifying relations between ⟨ne⟩s, also as proposed in MATE. An ⟨ante⟩ element includes one or more ⟨anchor⟩ element, one for each plausible antecedent of the current discourse entity (in this way, ambiguous cases can be marked). E.g., the anaphoric relation in (1) between the possessive pronoun with `ID ="ne09"` and the proper name with `ID ="ne07"` is marked as follows:

(2)
```
<ante current="ne09">
 <anchor ID="ne07" rel="ident" ... >
</ante>
```

(Discourse) Units

One difference between the annotation scheme we are using and the one discussed in (Poesio et al., 1999b) is that the problems we encountered trying to annotate centering information, proximity, and grammatical function (see also below) led us to mark up sentences and potential rhetorical units / centering theory utterances before marking up certain types of information about NPs such as grammatical function. The instructions for marking up units were in part derived from (Marcu, 1999); for each ⟨unit⟩, the following attributes were marked:

- utype: whether the unit is a main clause, a relative clause, appositive, a parenthetical, etc.
- verbed: whether the unit contains a verb or not.
- finite: for verbed units, whether the verb is finite or not.
- subject: for verbed units, whether they have a full subject, an empty subject (expletive, as in *there* sentences), or no subject (e.g., for infinitival clauses).

The agreement on identifying the boundaries of units was K = .9; the agreement on features was follows:

Attribute	K Value
utype	.76
verbed	.9
finite	.81
subject	.86

This part of the annotation has now been completed. The main difficulties we observed had to do with assigning an utterance type to parenthetical sentences.

NEs

After marking up units as discussed above, all NPs are marked up, together with a number of attributes. During our first round of experimentation we found that marking 'topics' in general was too difficult (K=.37), as was marking up thematic roles (K=.42); so although we haven't completely abandoned the idea of trying to annotate this information, in this second round we concentrated on improving the reliability for the other attributes. A few other attributes used in the previous scheme were dropped because they could be inferred automatically: among these are the feature disc specifying whether the discourse entity is discourse-new or discourse-old (redundant once antecedent information was marked up) and the feature cb used to mark whether the discourse entity is the current CB (Grosz et al., 1995) (which could be automatically derived from the information about grammatical function and units). We separated off information about the logical form type of an NP (quantifier, term, etc) from the information about genericity. Finally, new attributes were introduced to specify information which we found missing on the basis of our first evaluation: in particular, we decided to annotate information about the abstractness or concreteness of an object, and about its semantic plurality or atomicity. The revised list of information annotated for each NP includes:

- The output feature, cat, indicating the type of NP (e.g., bare-np, the-np, a-np).
- The other 'basic' syntactic features, num, per, and gen (for GENder).
- A feature gf specifying its grammatical function;
- The following semantic attributes:
  - ani: whether the object denoted is animate or inanimate
  - count: whether the object denoted is mass or count
  - lftype: one of `quant,term,pred,coord`
  - generic: whether the object denoted is a generic or specific reference
  - onto: whether the object denoted is concrete, an event, a temporal reference, or another abstract object
  - structure: whether the object denoted is atomic or not
- The following discourse attributes:
  - deix: whether the object is a deictic reference or not
  - loeb: whether the description used allows the reader to characterize the object as functional in the sense of Loebner (i.e., whether it denotes a single object, as in *the moon*, or at least a functional concept, like *father*)

A number of NP properties (e.g., familiarity) can be derived from the annotation of anaphoric information (below); in addition, a few properties of NPs are automatically derived from other sources of information - e.g., the type of layout element in which the NP occurs (in titles, bare-nps are often used) and whether a particular NP has uniquely distinguishing syntactic features in a given unit. All of these features can be annotated reliably, except for genericity; the results that we do have are as follows:

Attribute	**K Value**
cat	.9
gen	.89
num	.84
per	.9
gf	.85
ani	.91
count	.86
lftype	.82
onto	.80
structure	.82
deix	.81
loeb	.80

(One interesting point to note here is that agreement on lftype is actually quite high (90%), but because TERMs are so prevalent, chance agreement is also very high.)

We should point out that even though we reached a good level of agreement on all of these features, not in all cases it was easy to do so. The only features that are truly easy to annotate are NP type, person, and animacy. Good instructions are needed for gender, number, logical form, multiplicity, deixis, and uniqueness–e.g., for the case of gender one has to decide what to do with second person pronouns such as *you*, and for deixis the instructions have to specify what to do with objects that are not in the picture although appear to be visible. Finally, the count/mass distinction proved to be very difficult, as did the abstract / concrete distinction (e.g., are diseases abstract or concrete?). We did introduce a number of 'underspecified' values, but this did not lead to results as good as including in the instructions a number of examples (which suggests our scheme may not transport well to other applications).

Antecedent Information

Previous work, particularly in the context of the MUC initiative, suggested that while it's fairly easy to achieve agreement on identity relations, marking up bridging references is quite hard; this was confirmed, e.g., by (Poesio and Vieira, 1998). The only way to achieve a reasonable agreement on this type of annotation, and to contain somehow the annotators' work, is to limit the types of relations annotators are supposed to mark up, and specify priorities. We are currently experimenting with marking up only four types of relations, a subset of those proposed in the 'extended relations' version of the MATE scheme (Poesio et al., 1999a) (which, in turn, derived from Passonneau's DRAMA scheme (Passonneau, 1997): identity (`IDENT`), set membership (`ELEMENT`), subset (`SUBSET`), and 'generalized possession', including part-of relations.

In addition, given our interests we had to be quite strict about the choice of antecedent: whereas in MUC it is perfectly acceptable to mark an 'antecedent' which *follows* a given anaphoric expression, in order, e.g., to compute the CB of an utterance it is necessary to identify the *closest previous* antecedent.

As expected, we are achieving a rather good agreement on identity relations. In our most recent analysis (two annotators looking at the anaphoric relations between 200 NPs) we observed no real disagreements; 79.4% of these relations were marked up by both annotators; 12.8% by only one of them; and in 7.7% of the cases, one of the annotators marked up a closer antecedent than the other. On the other hand, only 22% of bridging references were marked in the same way by both annotators; although our current scheme does limit the disagreements on antecedents and relations (only 4.8% relations are actually marked differently) we still find that 73.17% of relations are marked by only one or the other annotator.

6 EVALUATION

In order to evaluate the complenetess of our schemes, we have been using the corpus annotated with the reliable features to build statistical models of the process of NP type determination - i.e., the process by which the value of `cat` is chosen on the basis of the values of the other features. We tried both the Maximum Entropy model (Berger et al., 1996) as implemented by Mikheev (Mikheev, 1998) and the CART model of decision tree construction (Breiman et al., 1984); the results below were obtained using CART. The models are evaluated by comparing the label it predicted on the basis of the features of a given NP with the actual value of `cat` for that NP, performing a 10-fold cross-validation.

The models discussed in (Poesio et al., 1999b) achieved a 70% accuracy, against a baseline of 22% (if the most common category, `BARE-NP`, is chosen every time.), training on a corpus of 3000 NPs. We are still in the process of evaluating the models built using our second corpus, but partial tests (trained on about 1,000 NPs) suggest that the using the new annotation scheme an accuracy of about 80% can be achieved.

The most complex problem to fix is that of `THIS-NP`s. The reason for the misclassification is that `THIS-NP`s are used in our texts not only to refer to pictures or parts of them, but also to refer to abstract objects introduced by the text, as in the following examples:

(3) a. *A great refinement among armorial signets was to reproduce not only the coat-of-arms but the correct tinctures; they were repeated in colour on the reverse side and the crystal would then be set in the gold bezel. Although the engraved surface could be used for impressions, the colours would not wear away. The signet-ring of Mary, Queen of Scots (beheaded in 1587) is probably the most interesting example of this type;*

b. *The upright secrétaire began to be a fashionable form around the mid-1700s, when letter-writing became a popular past-time. The marchands-merciers were quick to respond to this demand,* The problem is that such references are difficult to annotate reliably.

7 DISCUSSION

There are some pretty obvious omissions in the work done so far. Even if we only consider the task of NP type determination, there are a number of features whose impact we haven't been able to study so far, in some cases because they proved very hard to annotate. We already discussed two such examples, topichood and thematic roles; another potentially important source of information about the decision to pronominalize, rhetorical structure, is even harder to annotate. We would like to be able to annotate some types of scoping relations as well, especially the cases in which an NP is in the scope of negation as this may license the use of polarity-sensitive items such as *any*. Another important factor is the role of the information which the text planner has decided to realize: e.g., once the text planner has decided to generate both the proper name of discourse entity x, *Alphonse Mucha*, and the fact that x is a Czech painter, the decision to use the THE-NP *the Czech painter Alphonse Mucha* is more or less forced on us. And of course, nothing in the scheme discussed above allows us to study the conditions under which a generator may decide to produce a quantifier or a coordinated NPs.

Among the issues raised by this work, an important one is how much of the information that we annotated by hand could be automatically extracted. We believe that a lot of the syntactic information we rely on (⟨unit⟩ and ⟨ne⟩ identification, ⟨unit⟩ attributes, basic syntactic attributes of ⟨ne⟩) could be extracted automatically using recent advances in robust parsing; this would already cut down the amount of work considerably. The problem is what to do with semantic information: e.g., whether suitable approximations could be found.

Another important question is whether our characterization of NP realization is plausible. One possible objection is that NP type determination goes hand-in-hand with content determination, and the two problems can only be attacked simultaneously. The problem with this type of objection is that it's very difficult to study content determination. This is because of a more general problem with the methodology we are using: there is a mismatch between what a system knows and what an annotator may know about an object–i.e., between the features that a generation system may use and the features that can be annotated, and it's not clear this mismatch can be resolved.

For one thing, the need to choose features that can be annotated reliably imposes serious constraints: features that a generation system can easily set up by itself (e.g., the ILEX system keeps track of what it thinks the current topic is) can be difficult for two annotators to annotate in the same way. Second, some information that a generation system can use when deciding on the type of NP to generate may simply be impossible to annotate. For example, we already seen that the form of an NP often depends on how much information the system intends to communicate to the user about a given entity, or how much information the system believes the user has. In order to build a model of this decision process, we would need to specify for each NP how much information it conveys, and of what type; it's not at all clear that it will be feasible to do this by hand, except in domains in which the annotator knows everything that there is to know about a given object (see, e.g., Jordan's work on the COCONUT domain (Jordan, 1999)).

Conversely, some information that can be annotated - indeed, that is easy to annotate - may not be available to some systems. E.g., we do not know of any system with a lexicon rich enough to specify whether a given entry is functional or not. A solution in this case may be to develop algorithms to extract this information from an annotated corpus, or perhaps just using the syntactic distribution of the predicate as an indication (e.g., a predicate X occurring in a *the X of Y* construction may be functional).

In other words, we believe that the present work is only a first step towards developing an appropriate methodology for empirical investigation and evaluation of generation algorithms, which we nevertheless feel will become more and more necessary. But we believe that, already, this type of work can raise a number of interesting issues concerning semantic annotation and agreement on semantic judgments, which we hope to discuss at the workshop.

Acknowledgments

I wish to thank the other members of the GNOME project, without whom this work would not have been possible– in particular Hua Cheng, Kees van Deemter, Barbara di Eugenio, Renate Henschel, Rodger Kibble, and Rosemary Stevenson. The corpus is being annotated by Debbie De Jongh, Ben

Donaldson, Marisa Flecha-Garcia, Camilla Fraser, Michael Green, Shane Montague, Carol Rennie, and Claire Thomson. I also wish to thank Janet Hitzeman, Pam Jordan, Alistair Knott, Chris Mellish, Johanna Moore, and Jon Oberlander. The GNOME project is supported by the UK Research Council EPSRC, GR/L51126. Massimo Poesio is supported by an EPSRC Advanced Research Fellowship.

References

A. Berger, S. Della Pietra, and V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72.

L. Breiman, J. H. Friedman, R. A Olshen, and C. J. Stone. 1984. *Classification and Regression Trees*. Chapman and Hall.

J. Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

N. A. Chinchor and B. Sundheim. 1995. Message Understanding Conference (MUC) tests of discourse processing. In *Proc. AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 21–26, Stanford.

S. C. Garrod and A. J. Sanford. 1983. Topic dependent effects in language processing. In G. B. Flores D'Arcais and R. Jarvella, editors, *The Process of Language Comprehension*, pages 271–295. Wiley, Chichester.

B. J. Grosz, A. K. Joshi, and S. Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):202–225. (The paper originally appeared as an unpublished manuscript in 1986.).

J. K. Gundel, N. Hedberg, and R. Zacharski. 1993. Cognitive status and the form of referring expressions in discourse. *Language*, 69(2):274–307.

R. Henschel, J. Bateman, and C. Matthiessen. 1999. The solved part of NP generation. In R. Kibble and K. van Deemter, editors, *Proc. of the ESSLLI Workshop on Generating Nominals*, Utrecht.

J. Hitzeman, A. Black, P. Taylor, C. Mellish, and J. Oberlander. 1998. On the use of automatically generated discourse-level information in a concept-to-speech synthesis system. In *Proc. of the International Conference on Spoken Language Processing (ICSLP98),*, page Paper 591, Australia.

P. Jordan. 1999. An empirical study of the communicative goals impacting nominal expressions. In R. Kibble and K. van Deemter, editors, *Proc. of the ESSLLI workshop on The Generation of Nominal Expressions*, Utrecht. University of Utrecht, OTS.

S. Loebner. 1987. Definites. *Journal of Semantics*, 4:279–326.

D. Marcu. 1999. Instructions for manually annotating the discourse structures of texts. Unpublished manuscript, USC/ISI, May.

A. Mikheev. 1998. Feature lattices for maximum entropy modeling. In *Proc. of ACL-COLING*, pages 845–848, Montreal, CA.

J. Oberlander, M. O'Donnell, A. Knott, and C. Mellish. 1998. Conversation in the museum: Experiments in dynamic hypermedia with the intelligent labelling explorer. *New Review of Hypermedia and Multimedia*, 4:11–32.

R. Passonneau and D. Litman. 1993. Feasibility of automated discourse segmentation. In *Proceedings of 31st Annual Meeting of the ACL*.

R. Passonneau. 1997. Instructions for applying discourse reference annotation for multiple applications (DRAMA). Unpublished manuscript., December.

R. Passonneau. 1998. Interaction of discourse structure with explicitness of discourse anaphoric noun phrases. In M. A. Walker, A. K. Joshi, and E. F. Prince, editors, *Centering Theory in Discourse*, chapter 17, pages 327–358. Oxford University Press.

M. Poesio and R. Stevenson. To appear. *Salience: Computational Models and Psychological Evidence*. Cambridge University Press, Cambridge and New York.

M. Poesio and R. Vieira. 1998. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216, June. Also available as Research Paper CCS-RP-71, Centre for Cognitive Science.

M. Poesio, F. Bruneseaux, and L. Romary. 1999a. The MATE meta-scheme for coreference in dialogues in multiple languages. In M. Walker, editor, *Proc. of the ACL Workshop on Standards and Tools for Discourse Tagging*, pages 65–74.

M. Poesio, R. Henschel, J. Hitzeman, R. Kibble, S. Montague, and K. van Deemter. 1999b. Towards an annotation scheme for Noun Phrase generation. In B. Krenn H. Uszkoreit, T. Brants, editor, *Proc. of the EACL workshop on Linguistically Interpreted Corpora (LINC-99)*.

E. F. Prince. 1992. The ZPG letter: subjects, definiteness, and information status. In S. Thompson and W. Mann, editors, *Discourse description: diverse analyses of a fund-raising text*, pages 295–325. John Benjamins.

T. Reinhart. 1981. Pragmatics and linguistics: An analysis of sentence topics. *Philosophica*, 27(1). Also distributed by Indiana University Linguistics Club.

D. Scott, R. Power, and R. Evans. 1998. Generation as a solution to its own problem. In *Proc. of the 9th International Workshop on Natural Language Generation*, Niagara-on-the-Lake, CA.

An Environment for Extracting Resolution Rules of Zero Pronouns from Corpora

Hiromi Nakaiwa
NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Souraku-gun, Kyoto 619-0237 Japan
nakaiwa@cslab.kecl.ntt.co.jp

Abstract

This paper proposes a practical integrated environment for extracting rules for the anaphora resolution of zero pronouns from monolingual and/or bilingual corpora. This method takes into account the practical situation for making resolution rules of zero pronouns in specific domain texts; the types of usable corpora (monolingual and/or bilingual) for examining the extraction of resolution rules have been changed depending on the type of NLP system using extracted resolution rules. The extraction processes of resolution rules in the environment are classified into five component tasks: (1) Zero Pronoun Identification, (2) Antecedent Annotation, (3) Rejection of Sentences Unsuitable for Rule Extraction, (4) Rule Extraction, and (5) Extracted Rule Application and Modification. An automatic process and/or a manual process with a user friendly human interface can be used to achieve each component task. This environment was implemented in the Japanese-to-English machine translation system, **ALT-J/E**, for Japanese zero pronoun resolution.

1 Introduction

In natural languages, elements that can be easily deduced by a reader are frequently omitted from expressions in texts (Kuno, 1978). This phenomenon causes considerable problems in NLP systems such as MT, text summarization and text retrieval. In particular, the subject and object are often omitted in Japanese, whereas they are normally obligatory in English[1]. In Japanese-to-English machine translation systems, therefore, it is necessary to identify case elements omitted from the original Japanese ("zero pronouns") for their accurate translation into English expressions.

Several algorithms have been proposed with regard to this problem (Kameyama, 1986; Yoshimoto, 1988; Walker et al., 1990; Dohsaka, 1994). However, it is not possible to apply these methods directly to a practical machine translation system because of their low precision of resolution and the large volume of knowledge required.

To overcome these kinds of problems, several methods have been proposed (Nakaiwa and Ikehara, 1992; Nakaiwa and Ikehara, 1995; Nakaiwa and Ikehara, 1996). The focus of these methods is on applications for a practical machine translation system with an unlimited translation target area.

With these methods, however, it is necessary to make resolution rules for zero pronouns by hand. Unfortunately, it takes a lot of time and effort for the experts of the NLP system to make these rules robust and with wide coverage. Furthermore, resolution rules often have to be made depending on the target domain of the documents, and this also requires the time-consuming labor of experts. Because of these problems, there is a need for an effective and efficient method of making resolution rules for zero pronouns.

Typical methods for this purpose include extracting resolution rules for zero pronouns from monolingual corpora (Nasukawa, 1996; Murata and Nagao, 1997), from bilingual corpora (Nakaiwa, 1997a; Nakaiwa, 1997b), and from monolingual corpora with tags for antecedents of zero pronouns (Aone and Bennett, 1995; Yamamoto and Sumita, 1998).

Monolingual corpora are relatively easy to collect. Methods using monolingual corpora, however, have difficulties in extracting resolution rules of zero pronouns whose referents are normally unexpressed in Japanese.

Methods using sentence-aligned bilingual corpora are better than those using monolingual corpora. This is particularly so with a bilingual corpus of dissimilar languages such as Japanese and English whose language families are so different and where the distributions of zero pronouns are also quite different. However, bilingual corpora are relatively difficult to collect,

[1] For example, there are 93 omitted case elements in 102 sentences in 30 newspaper articles which have to be explicitly translated into English.

especially sentence-aligned corpora.

With methods using monolingual corpora with antecedent tags, it is possible to efficiently make effective resolution rules by relying on the annotated information. However, there are only a few corpora with antecedent tags for zero pronouns. The standardization for annotating zero pronouns and their antecedents is still ongoing (Hasida, 2000). Consequently, in actual situations, analysts who want to make resolution rules for zero pronouns also have to laboriously annotate antecedent tags to zero pronouns in the corpus by hand, as previously mentioned. Therefore, an annotation tool for the antecedents of zero pronouns in the texts (Aone and Bennett, 1994) is needed for the effective addition of tags to zero pronouns.

To create resolution rules of zero pronouns in a text of a specific domain, we commonly use only monolingual corpora in the specific domain without antecedent tags for zero pronouns. Accordingly, analysts annotate tags to the antecedents of every zero pronoun in the corpus to make effective resolution rules. However, to accomplish this in machine translation, it is also possible to use bilingual corpora in the specific domain, such as a former version of a text that has already been translated or bilingual corpora used for translation memory systems. In this case, methods that automatically extract the resolution rules of zero pronouns from bilingual corpora (Nakaiwa, 1997a; Nakaiwa, 1997b) can be used. An automatic extraction process, however, cannot make perfect rule sets. Therefore, the automatically extracted rules have to be confirmed by human interaction before adding the rule set used in anaphora resolution in NLP systems if highly reliable rules such as domain-independent default rules are required. Furthermore, the human interaction must take into account the efficiency of acquiring resolution rules from both monolingual and bilingual corpora.

Considering these practical conditions for extracting the resolution rules of zero pronouns, this paper proposes a practical integrated tool capable of extracting rules for the anaphora resolution of zero pronouns from monolingual and/or bilingual corpora.

2 Component Tasks of Resolution Rule Extraction of Zero Pronouns

We classify the subtasks for extracting resolution rules from corpora into the following five component tasks: (1) Zero Pronoun Identification, (2) Antecedent Annotation, (3) Rejection of Sentences for Rule Extraction, (4) Rule Extraction, and (5) Extracted Rule Application and Modification.

2.1 Zero Pronoun Identification

The zero pronoun identification process identifies zero pronouns that must be resolved in an NLP system using extracted resolution rules. For example, Japanese, which is a free word-order language, often has no explicit cue helpful in determining obligatory case elements. Therefore, in this language, the identification of zero pronouns in the corpus is also important for extracting resolution rules. Furthermore, depending on the NLP system, the zero pronouns that must be resolved are different. For example, MT systems only need to resolve zero pronouns that must be explicitly translated into the target. In a Japanese sentence (1), the subject (*ga*-case) is not expressed in Japanese but becomes optional when translated into English, because it is possible to translate this by using the expression, "Zoos raise lions.".

(1) (*ϕ-ga*) *doubutsuen-de* *raion-o* *kau.*
ZOO-AT lion-OBJ keep
Zoos raise lions.

Therefore, in the zero pronoun identification process, the analysis results of the NLP system must be taken into account.

Zero pronoun identification in monolingual corpora only relies on the analysis results of the NLP system. In bilingual corpora, however, the translation equivalent of zero pronouns is also usable as a trigger for determining zero pronouns that must be resolved.

2.2 Antecedent Annotation

The antecedent annotation process identifies antecedents of zero pronouns that need to be resolved. In monolingual corpora, analysts must basically annotate antecedents of zero pronouns manually. However, even in the manual process, the following factors must be taken into account.

- Zero pronouns with the same syntactic and semantic features (such as modal expressions, the meaning of verbs, and conjunctions) around them in the corpus should be grouped and displayed at the same time when their antecedents are annotated.
  Zero pronouns with the same features tend to have the same type of antecedents because the features become key factors in

determining their antecedents. Therefore, analysts can judge antecedents for zero pronouns with the same features easily and efficiently.

- Antecedent candidates of zero pronouns should be easy to select from elements in the text or deictic elements outside the text.
  There are three types of possible antecedent candidates for each zero pronoun: candidates in the same sentence (intrasentential), candidates in another sentence in the text (intersentential), and candidates that are not explicitly expressed in the text (deictic). Intrasentential and intersentential antecedent candidates are actually expressed in the text. Their conditions in the resolution rules involve their syntactic positions and/or sentential relationships such as distance, rhetorical relation, and relative relation in the discourse structure (e.g., a candidate in the title of a section and a zero pronoun in a sentence in the section) (Nakaiwa and Ikehara, 1992; Nakaiwa and Ikehara, 1995). Therefore, by grouping intra- and intersentential candidates with the same syntactic position and sentential relationship in the text, and by showing the same types of candidates for zero pronouns at the same time, we can select the actual antecedent easily and efficiently. Among deictic antecedent candidates, the antecedents tend to be limited elements such as *writer/speaker* or *reader/hearer* (Nakaiwa and Ikehara, 1996). Therefore, listing the possible antecedent candidates before the annotation process and selecting the actual antecedent from the possible antecedent candidate list make the annotation process of deictic antecedents for zero pronouns much easier and more efficient.

In the case of bilingual corpora, in addition to the manual process used for monolingual corpora, the translation of a sentence with zero pronoun can be used to determine the antecedent of the zero pronoun. For example, in Japanese and English bilingual corpora, the subject and object are often omitted in Japanese, whereas they are normally obligatory in English. Therefore, by aligning zero pronouns in Japanese and their translation equivalents in English, antecedent of zero pronouns can be automatically identified (Nakaiwa, 1997b).

2.3 Rejection of Sentences Unsuitable for Rule Extraction

The following types of sentences with zero pronouns and/or antecedents in the corpus are not suitable as the source sentences for extracting rules.

(a) Sentences in which the analysis made errors in identifying the predicate, e.g., an adverbial expression, modal expression, or postpositional phrase as a predicate. This type of error identifies zero pronouns erroneously.

(b) Translation-equivalent sentences in bilingual corpora that were freely translated by a human. Here, it is very difficult to identify the translation equivalents of the zero pronouns within the translation-equivalent sentence in the automatic identification process.

The problematic sentences with zero pronouns and/or erroneous zero pronouns in type (a) have to be annotated as 'unsuitable' before extracting rules from sentences with zero pronouns in the corpus. The antecedents of zero pronouns in problematic sentences in type (b) have to be manually annotated even with bilingual corpora.

2.4 Rule Extraction

The rule extraction process extracts resolution rules of zero pronouns with antecedent tags in the corpus. In this process, syntactic and semantic features around zero pronouns and around their annotated antecedents are used as a condition in the resolution rules.

There are two way to extract rules:

(a) Automatic Extraction
In this process, resolution rules can be automatically extracted from zero pronouns with antecedent tags (Section 2.2) and syntactic and semantic features around zero pronouns and their annotated antecedents by a machine learning technique (Aone and Bennett, 1995; Yamamoto and Sumita, 1998) or by statistical processing (Nakaiwa, 1997a).

(a) Manual Extraction
In this process, zero pronouns are grouped depending on their syntactic position, their annotated antecedent, and the syntactic and semantic features around the zero pronouns. Resolution rules for the grouped zero pronouns are extracted by examining how many correct antecedents for the zero

pronouns can be covered under the same features.

2.5 Extracted Rules Application and Modification

The extracted rules in section 2.4 are used by the NLP system in the resolution of zero pronouns in sentences in the corpus used for the rule extraction. Considering the results of the application of extracted rules for zero pronouns, we examine the suitability of rules for the corpus. If there are some problems in the resolution rules, the problematic rules and/or the priorities of the rules are modified.

The rule set with the modified rules is again used by the NLP system for the same corpus, and the suitability of rule modifications is checked in the same manner. The rule modification and re-application for zero pronouns within the corpus are iterated until reasonable rules are extracted.

3 Implemented Architecture for Extracting Resolution Rules of Zero Pronouns from Corpora

Considering the five components described in section 2, we have implemented an architecture for automatically and/or manually extracting resolution rules for Japanese zero pronouns from Japanese and English bilingual corpora and/or Japanese monolingual corpora. Figure 1 shows an overview of the system. In the first step, the Japanese and English sentences in the bilingual corpus and/or the Japanese sentences in the monolingual corpus are separately analyzed by Japanese and English parsers. In the next step, the antecedents of zero pronouns within the Japanese sentences in the corpus are identified automatically from Japanese and English analysis results in the bilingual corpus. From the monolingual corpus, however, only the Japanese analysis results with the syntactic position of zero pronouns are extracted. The Japanese analysis results with/without antecedent tags for zero pronouns are stored as 'Japanese Corpus with Antecedent Tags' as shown in the figure. Each zero pronoun in the corpus is manually examined in order to annotate the correct antecedent tags, if required. From the annotated information, resolution rules for zero pronouns are extracted manually or automatically. Manual extraction is preferable for acquiring reliable rules, but requires a high cost. In contrast, the automatic extraction process has a high possibility of extracting problematic rules. Such types of automatically extracted rules require human checks and modifications for extracting reliable rule sets.

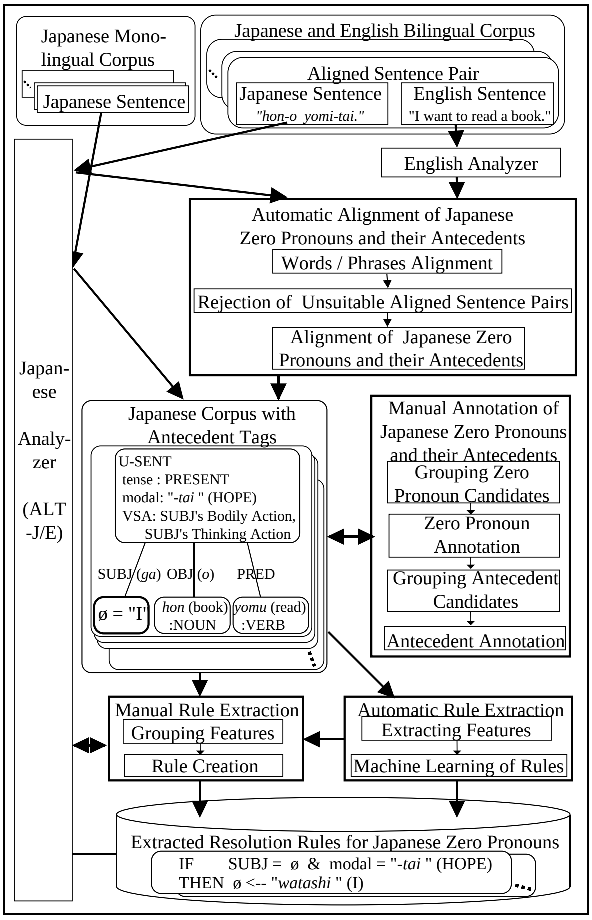

Japanese Mono-lingual Corpus
Japanese Sentence
Japanese and English Bilingual Corpus
Aligned Sentence Pair
Japanese Sentence
"hon-o yomi-tai."
English Sentence
"I want to read a book."
English Analyzer
Automatic Alignment of Japanese Zero Pronouns and their Antecedents
Words / Phrases Alignment
Rejection of Unsuitable Aligned Sentence Pairs
Alignment of Japanese Zero Pronouns and their Antecedents
Japan-ese Analy-zer (ALT -J/E)
Japanese Corpus with Antecedent Tags
U-SENT
tense : PRESENT
modal: "-tai" (HOPE)
VSA: SUBJ's Bodily Action, SUBJ's Thinking Action
SUBJ (ga)
OBJ (o)
PRED
ø = "I"
hon (book) :NOUN
yomu (read) :VERB
Manual Annotation of Japanese Zero Pronouns and their Antecedents
Grouping Zero Pronoun Candidates
Zero Pronoun Annotation
Grouping Antecedent Candidates
Antecedent Annotation
Manual Rule Extraction
Grouping Features
Rule Creation
Automatic Rule Extraction
Extracting Features
Machine Learning of Rules
Extracted Resolution Rules for Japanese Zero Pronouns
IF SUBJ = ø & modal = "-tai" (HOPE)
THEN ø <-- "watashi" (I)

Figure 1: Process for Extraction of Resolution Rules for Japanese Zero Pronouns

In the next step, the extracted resolution rules are used for the anaphora resolution of Japanese zero pronouns in the corpora by the Japanese analyzer. The same sentences in the monolingual and/or bilingual corpora are inputted to the system and resolution rules are again extracted and modified for the Japanese zero pronouns. These processes are repeated until the system cannot extract any more resolution rules for the Japanese zero pronouns in the corpora.

This method has been implemented in a Japanese-to-English machine translation system, **ALT-J/E** (Ikehara and et al., 1991). The system in Figure 1 can extract English translation equivalents of Japanese zero pronouns from aligned sentence pairs. Accordingly, the results can also be used to extract rules for translating Japanese zero pronouns into English in a Japanese-to-English machine translation system. For efficient human interaction in the manual process, we use the interface of a widely used WWW browser, such as Netscape Navigator or Internet Explorer.

In the following subsections, we describe the details of automatically and manually extracting resolution rules for the Japanese zero pronouns in the corpora.

3.1 Analysis of Japanese and English Sentences

Japanese sentences and English sentences in the corpora are analyzed in the following manner.

3.1.1 Analysis of Japanese Sentences

Japanese sentences are analyzed with the morphological, syntactic, and semantic analyzers of Japanese in **ALT-J/E** (Ikehara and et al., 1991). The syntactic and semantic structure of the Japanese sentence is first created. The Japanese structure is used for the automatic translation into English in **ALT-J/E**. The Japanese structure, therefore, includes the syntactic positions of the Japanese zero pronouns, which must be translated into English, and the semantic constraints for the Japanese zero pronouns forced by the verb within the Japanese sentence. When a zero pronoun is resolved by a rule, a determined antecedent and an ID of the applied rule for each zero pronoun are also annotated. This information is used to judge whether existing rules will resolve zero pronouns successfully, and which zero pronouns require further resolution rules.

For example, from the Japanese sentence in the aligned sentence pair (2) in Figure 1, the following syntactic and semantic structure is created.

(2) (*ϕ-ga*) *hon-o* *yomi-tai*
book-OBJ read-WANT-TO
I want to read a book.

(3) Syntactic and Semantic Structure of Japanese Sentence (2)

```
┌U_SENT-1                                                   ┐
│Tense     PRESENT, PERFECTIVE ASPECT                        │
│Modal     tai (HOPE)                                        │
│VSA       SUBJECT'S HUMAN ACTION,                           │
│          SUBJECT'S THINKING ACTION                         │
│PRED-1    [main verb  yomu "read"]                          │
│          ┌case rel.  OBJECT "o"┐                           │
│CASE-1    │NP-1       hon "book"│                           │
│          └                     ┘                           │
│          ┌case rel.                 SUBJ ┐                 │
│CASE-2    │NP-2                      ϕ-1  │                 │
│          │semantic constraints      HUMAN│                 │
└          └                               ┘                 ┘
```

3.1.2 Analysis of English Sentences

English sentences are analyzed by Brill's English Tagger (Brill, 1992) and the Link Grammar Parser (Sleator and Temperley, 1991). Next, the syntactic structure is converted into a partial syntactic structure, which is similar to the internal English structure of **ALT-J/E**.

For example, from the English sentence in aligned sentence pair (2), the following partial syntactic structure is created.

(4) Partial syntactic structure of an English Sentence (2)

```
┌U_SENT-1                                                           ┐
│          ┌"want"  VERB, SING., PRESENT.┐                          │
│PRED-1    │"to"    TO                   │                          │
│          └"read"  VERB, BASE FORM      ┘                          │
│          ┌case rel.  SUBJECT                      ┐               │
│CASE-1    │NP-1       ["I" : PERSONAL PRONOUN]     │               │
│          └                                        ┘               │
│          ┌case rel.  DIRECT OBJECT                     ┐          │
│          │           ┌"a"     DETERMINER          ┐      │          │
│CASE-2    │NP-2       │"book"  NOUN,               │      │          │
│          │           │        SINGULAR OR MASS    │      │          │
└          └           └                            ┘      ┘          ┘
```

3.2 Automatic Alignment of Japanese Zero Pronouns and their Antecedents[2]

From the analysis results of Japanese and English aligned sentence pairs, the system extracts pairs of Japanese words/phrases and their English equivalent words/phrases by comparing the two structures. Then, based on the discussion in section 2.3, aligned sentence pairs not suitable for the extraction of resolution rules for Japanese zero pronouns are automatically identified if any of the following conditions are met.

- There is a difference between the number of clauses whose Japanese verb is not aligned with an English noun, within the Japanese analysis result, and the number of clauses within the English analysis result.
- The MT system fails to translate some words.
- The English Parser is unable to make a full syntactic structure.

Next, the Japanese zero pronouns in the Japanese sentences and the translation equivalents of their antecedents in English sentences are extracted using 10 hand-developed alignment rules.

For example, from the zero pronoun in the *ga*-case (subject) in the Japanese sentence in

[2] This process is implemented by using the alignment method proposed by Nakaiwa (Nakaiwa, 1997b).

aligned sentence pair (2), its antecedent is automatically determined as the subject in the English sentence (“*I*’), as shown in the ‘Japanese Corpus with Antecedent Tags’ block in Figure 1.

3.3 Manual Annotation of Japanese Zero Pronouns and their Antecedents

With Japanese monolingual corpora, an analyst who wants to make resolution rules for Japanese zero pronouns in the corpora must annotate their antecedents by hand. To achieve an efficient annotation process, we have developed a tool for annotating antecedents of Japanese zero pronouns in Japanese sentences within the corpora. This process uses the analysis results of Japanese sentences (section 3.1.1). The details of this process are described in the following sections.

3.3.1 Identifying Zero Pronouns

According to the results of the Japanese analysis in a Japanese-to-English MT system, the zero pronouns that must be explicitly translated in English are explicitly annotated in the syntactic and semantic structure of the inputted Japanese sentences (e.g., example (3)). However, as we discussed in section 2.3, the sentence analysis error causes erroneous zero pronouns. Therefore, the analyst must annotate whether the zero pronoun candidates in the Japanese analysis result are actually zero pronouns or not. For efficiency, the Japanese analysis results are grouped based on whether the same features are around zero pronoun candidates as follows.

(1) syntactic position of zero pronoun candidates (e.g., *ga-case* (Subject), *o-case* (Direct Object)).

(2) syntactic and semantic structure around zero pronoun candidates (e.g., the types of conjunctions, verbal semantic attributes, and the types of modal expressions in unit sentences with zero pronouns candidates).

Figure 2 shows an example of the display after grouping zero pronoun candidates according to their syntactic positions. As shown in the figure, N1 (*ga*-case) is the most common syntactic position of zero pronouns in the corpus (866 instances in 724 sentences), and N2 (*o*-case) is the next most common (125 instances in 116 sentences).

3.3.2 Annotating Antecedents of Zero Pronouns

After identifying zero pronouns in the Japanese sentences, an antecedent for each zero pronoun is annotated. As with the process of zero pronoun identification, zero pronouns are grouped based on the presence of the same characteristics around the zero pronouns. To efficiently annotate the antecedents of zero pronouns, we also group intrasentential and intersentential anaphora candidates according to the characteristics around the candidates as follows:

(1) syntactic position of an antecedent candidate

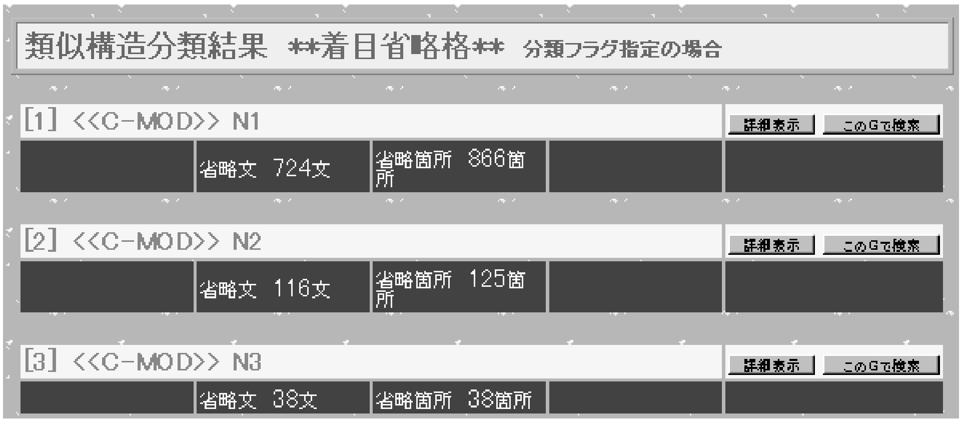
(2) syntactic and semantic structures around an antecedent candidate

(3) syntactic relationship between a unit sentence with a zero pronoun and a unit sentence with an antecedent candidate in the same sentence (intrasentential) (e.g., a unit sentence with a zero pronoun is directly connected to another unit sentence by a conjunction)

(4) discourse structural relationship (or distances) between a sentence with a zero pronoun and a sentence with an antecedent candidate (intersentential) (e.g., a sentence with a zero pronoun is the next sentence following a sentence with an antecedent candidate)

For (3), the syntactic structures of unit sentences with zero pronouns are classified, and the sentences with the same types of syntactic structures are examined. For (4), typical antecedent candidate relationships for the target corpus are selected in advance. For example, in newspaper articles, the first sentence of an article often contains the antecedent of a zero pronoun in another sentence in the article (Nakaiwa and Ikehara, 1992). The relationship between sentences should be selected depending on the target domain to achieve an efficient annotating process. An analyst annotates deictic antecedents of zero pronouns by first selecting typical antecedent candidates such as “I/we”, “you”, or “it” in advance and then selecting the diectic antecedent of a zero pronoun from them. After an antecedent for a zero pronoun is annotated, other antecedent candidates for the zero pronoun are displayed as “negative candidates” in the display of the grouping result.

By grouping antecedent candidates having the same characteristics, analysts can efficiently



類似構造分類結果 **着目省略格** 分類フラグ指定の場合

[1] <<C-MOD>> N1

省略文 724文 省略箇所 866箇所

[2] <<C-MOD>> N2

省略文 116文 省略箇所 125箇所

[3] <<C-MOD>> N3

省略文 38文 省略箇所 38箇所

Figure 2: Display of Grouping Result of Zero Pronouns Candidates according to their Syntactic Positions

annotate antecedents of zero pronouns by referring to the antecedent candidates within the same type of context. Furthermore, by annotating antecedents from the context with high frequency to low frequency, an analyst can efficiently annotate antecedents in the early stage of the annotating process.

3.4 Automatic Extraction of Resolution Rules

Syntactic and semantic features around zero pronouns and their antecedents are extracted from Japanese sentences with Japanese analysis results and with tags for zero pronouns and their antecedents. The following features, the effects of which were discussed in Nakaiwa (1992;1995;1996) are used as conditions of extracted resolution rules.

- verbal semantic attributes (Nakaiwa et al., 1994)
- type of modal expression (Kawai, 1987)
- type of conjunction between a unit sentence with a zero pronoun and a unit sentence with its antecedent
- syntactic relationship between a unit sentence with a zero pronoun and a unit sentence with its antecedent (intrasentential)
- discourse structural relationship (or distance) between a sentence with a zero pronoun and a sentence with its antecedent (intersentential)

Rules are automatically extracted by a decision tree learning program, C5.0 (Quinlan., 1998). Extracted rules are converted to the rule format used in **ALT-J/E**.

3.5 Manual Extraction of Resolution Rules

For the extraction of more reliable resolution rules with human interaction, a manual rule extraction process from Japanese sentences using Japanese analysis results and tags for zero pronouns and their antecedents is implemented in the system. In the same manner as in section 3.3, the five types of features around zero pronouns and their antecedents used in section 3.4 are grouped and sorted by the frequencies of the grouped items. Therefore, wide coverage rules are efficiently extracted in the early stage of the extraction process. This is also effective for rule extraction by taking into account zero pronouns with the same types of context. The reliability of the extracted rules is also examined in this stage by calculating the number of applied zero pronouns for each rule and the number of successfully resolved zero pronouns by referring to antecedent tags for zero pronouns. Before the extracted rules are added to the rule set used in **ALT-J/E**, inclusion relationships between rules are examined and the priorities of extracted rules within the rule set are set.

4 Preliminary Evaluation

The performance of the automatic extraction process from aligned sentence pairs has been reported in (Nakaiwa, 1997a; Nakaiwa, 1997b). According to the evaluation result on the automatic alignment of Japanese zero pronouns and the English equivalents of their antecedents, 98.4% of all pairs were automatically aligned correctly in the training data and 94% of all pairs in unseen test data. Furthermore, according to their evaluation of extracted rules for zero pronouns with deictic references, those rules created automatically from sentence pairs correctly resolved 99.0% of the zero pronouns in the training data and 85.0% of the zero pronouns in an unseen test data. Therefore, we only evaluate the manual extraction process from Japanese monolingual corpora. The effectiveness of the proposed method of manual rule extraction highly relies on their grouping function. Therefore, in this evaluation, we examine the effectiveness of the manual rule extraction with or without the grouping function.

4.1 Evaluation Method of Manual Rule Extraction

The effectiveness and efficiency of manual rule extraction is examined by extracting rules from 3719 Japanese sentences in a test set for evaluating Japanese-to-English MT system (Ikehara et al., 1994). An analyst who is an expert of zero pronoun resolution in **ALT-J/E**, extracts resolution rules using the implemented system in section 3, which is installed in SUN Sparc Enterprise 3000, in the following manner.

A. Extraction using the Grouping Function
The grouping, annotation and extraction are conducted as follows.

Step 1 Zero pronoun candidates are grouped according to their syntactic positions; the candidates in the most common syntactic position, N1 (*ga*-case) are selected: 866 instances in 724 sentences (Figure 2)

Step 2 Selected candidates are grouped again according to their syntactic structure; the candidates in the most common syntactic structure, where a unit sentence with a zero pronoun is directly connected with another unit sentence by a conjunction, are selected: 315 instances

Step 3 Zero pronouns are annotated for the selected candidates: 285 out of 315 instances

Step 4 Antecedents of selected zero pronouns are annotated for 285 zero pronouns after grouping the type of conjunctions.

Step 5 Five rules are extracted from 285 zero pronouns and the required time for making the rules is recorded.

B. Extraction without using the Grouping Function
Rules are extracted from sentences with zero pronoun candidates one by one without using the grouping function in the time it takes to make the five rules in test A.

The results of two tests are compared by examining how fast the antecedents of zero pronouns can be efficiently annotated and how many zero pronouns can be successfully resolved by using extracted rules.

4.2 Evaluation Result

Table 1 shows the results of the evaluation. As shown in this table, zero pronouns and their antecedents were efficiently annotated in test A (1.1 min/item and 1.7 min/item using the grouping function (test A), and 2.5 min/item and 2.0 min/item without using the grouping function (test B), respectively). Furthermore, the rule extraction time and its application and evaluation time were also shorter in test A than in test B (1.1 min/item and 2.2 min/item in test A, and 6.0 min/item and 10.0 min./item in test B, respectively). This result indicates that grouping results with annotated information is helpful for making rules with wide coverage by taking into account the estimated result of an extracting rule for zero pronouns that will be applied. Regarding the quality of extracted rules, test B extracted better rules than test A (93 % in test A and 100 % in test B). However, the five problematic zero pronouns in test A were already noticed by the analyst at the rule evaluation step. Therefore, new rules for the zero pronouns with a detailed condition will be extracted easily by referring to this result.

Table 1: Required Time and Accuracy of Manually Extracted Rules (required time per zero pronoun shown in parentheses)

Grouping Function		used (A)	unused (B)
# of Extracted Rules		5	51
Required Time [min]	Zero Pron. Identification	332 (1.1)	128 (2.5)
	Antecedent Identification	482 (1.7)	102 (2.0)
	Rule Extraction	77 (1.1)	306 (6.0)
	Rule Application and Evaluation	158 (2.2)	510 (10.0)
	Total	1049 (6.0)	1046 (20.5)
# of Applied Zero Pron.		72	51
# of Correctly Resolved Zero Pron.		67 (93%)	51 (100%)

5 Conclusions

This paper proposed a practical integrated tool for extracting rules for the anaphora resolution of zero pronouns from monolingual and/or bilingual corpora. According to the preliminary evaluation of the manual rule extraction process, antecedent tags for zero pronouns can be efficiently annotated and rules are efficiently extracted from Japanese monolingual corpora by using the tool's grouping function. In the future, we will examine the effectiveness of the proposed method for both monolingual and bilingual corpora. We will also examine the most effective combined strategies for the extraction of resolution rules by using both automatic and manual processes.

References

Chinatsu Aone and Scott W. Bennett. 1994. Discourse tagging tool and discourse-tagged multilingual corpora. In *Proc. of the Internationa1 Workshop on Sharable Natural Language Resources*, pages 71–77.

Chinatsu Aone and Scott W. Bennett. 1995. Automated acquisition of anaphora resolution strategies. In *Working Notes of AAAI Spring Symposium Series, Empirical Methods in Discourse Interpretation and Generation*, pages 1–7.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proc. of ANLP-92*, pages 152–155.

Kohji Dohsaka. 1994. Identifying the referents of Japanese zero-pronouns based on pragmatic condition interpretation. *Transaction of IPSJ*, 35(10):34–40. In Japanese.

Koiti Hasida. 2000. Global document annotation (GDA). `http://www.etl.go.jp/etl/nl/GDA/`.

Satoru Ikehara and Satoshi Shirai et al. 1991. Toward MT system without pre-editing – effects of new methods in **ALT-J/E** –. In *Proc. of MT Summit III*, pages 101–106. (`http://xxx.lanl.gov/abs/cmp-lg/9510008`).

Satoru Ikehara, Satoshi Shirai, and Kentaro Ogura. 1994. Criteria for evaluating the linguistic quality of Japanese-to-English machine translation. *Journal of JSAI*, 9(5):569–579.

Megumi Kameyama. 1986. A property-sharing constraint in centering. In *Proc. of the 24th Annual Meeting of ACL*, pages 200–206.

Atsuo Kawai. 1987. Modality, tense and aspect in Japanese-to-English translation system ALT-J/E. In *Proc. of the 34th Annual Conv. of IPSJ*, pages 1245–1246. In Japanese.

Susumu Kuno. 1978. *Danwa no Bunpoo.* Taishukan Publ. Co., Tokyo, Japan. In Japanese.

Masaaki Murata and Makoto Nagao. 1997. An estimation of referents of pronouns in Japanese sentences using examples and surface expressions. *Journal of Natural Language Processing*, 4(1):87–109.

Hiromi Nakaiwa and Satoru Ikehara. 1992. Zero pronoun resolution in a Japanese-to-English machine translation system by using verbal semantic attributes. In *Proc. of ANLP-92*, pages 201–208.

Hiromi Nakaiwa and Satoru Ikehara. 1995. Intrasentential resolution of Japanese zero pronouns in a machine translation system using semantic and pragmatic constraints. In *Proc. of TMI-95*, pages 96–105.

Hiromi Nakaiwa and Satoru Ikehara. 1996. Anaphora resolution of Japanese zero pronouns with deictic reference. In *Proc. of COLING-96*, pages 812–817.

Hiromi Nakaiwa, Akio Yokoo, and Satoru Ikehara. 1994. A system of verbal semantic attributes focused on the syntactic correspondence between Japanese and English. In *Proc. of COLING-94*, pages 672–678.

Hiromi Nakaiwa. 1997a. Automatic extraction of rules for anaphora resolution of Japanese zero pronouns from aligned sentence pairs. In *Proc. of ACL-97/EACL-97 Workshop on Operational Factors in Practical, Robust, Anaphora Resolution for Unrestricted Texts*, pages 22–29. ACL.

Hiromi Nakaiwa. 1997b. Automatic identification of zero pronouns and their antecedents within aligned sentence pairs. In *Proc. of the 5th WVLC*, pages 127–141.

Tetsuya Nasukawa. 1996. Full-text processing: Improving a practical NLP system based on surface information within the context. In *Proc. of COLING-96*, pages 824–829.

J. Ross Quinlan. 1998. `http://www.rulequest.com/`.

Daniel Sleator and Davy Temperley. 1991. Parsing English with a link grammar. *Carnegie Mellon University Computer Science Technical Report*, pages CMU–CS–91–196.

Marilyn Walker, Masayo Iida, and Sharon Cote. 1990. Centering in Japanese discourse. In *Proc. of COLING-90*, pages 1–6.

Kazuhide Yamamoto and Eiichiro Sumita. 1998. Feasibility study for ellipsis resolution in dialogues by machine-learning technique. In *Proc. of COLING-ACL-98*, pages 1428–1434.

Kei Yoshimoto. 1988. Identifying zero pronouns in Japanese dialogue. In *Proc. of COLING-88*, pages 779–784.

DISCOURSE STRUCTURE ANALYSIS FOR NEWS VIDEO

Yasuhiko Watanabe[†] *Yoshihiro Okada*[†] *Sadao Kurohashi*[‡] *Eiichi Iwanari*[†]

[†] Dept. of Electronics and Informatics, Ryukoku University, Seta, Otsu, Shiga, Japan
[‡]Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo, Kyoto, Japan
watanabe@rins.ryukoku.ac.jp

ABSTRACT

Various kinds of video recordings have discourse structures. Therefore, it is important to determine how video segments are combined and what kind of coherence relations they are connected with. In this paper, we propose a method for estimating the discourse structure of video news reports by analyzing the discourse structure of their transcripts.

1. INTRODUCTION

A large number of studies have been made on video analysis, especially segmentation, feature extraction, indexing, and classification. On the other hand, little attention has been given to the discourse structure (DS) of video data.

Various kinds of video recordings, such as dramas, documentaries, news reports, and sports castings, have discourse structures. In other words, each video segment of these video recordings is related to previous ones by some kind of relation (coherence relation) which determines the role of the video segments in discourse. For this reason, it is important to determine how video segments are combined and what kind of coherence relations they are connected with. In addition, Nagao et.al proposed a method for multimedia data summarization using GDA tags [Nagao 00]. However, the cost of making GDA tagged data is great. Our method will be helpful in reducing the annotation cost.

In this paper, we propose a method for estimating the discourse structure of video news reports. Coherence relations between video segments are estimated in the following way:

1. a video news article is segmented into shots by using DCT components,
2. consecutive shots are merged by using speech information, and
3. coherence relations are estimated by using three kinds of clues in the transcript of the news video:
   - clue expressions indicating a certain relation,
   - occurrence of identical/synonymous words/ phrases in topic chaining or topic-dominant chaining relation, and
   - similarity between two sentences in list or contrast relation.

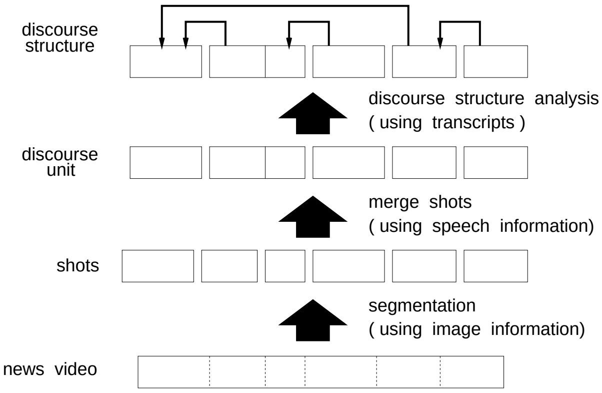
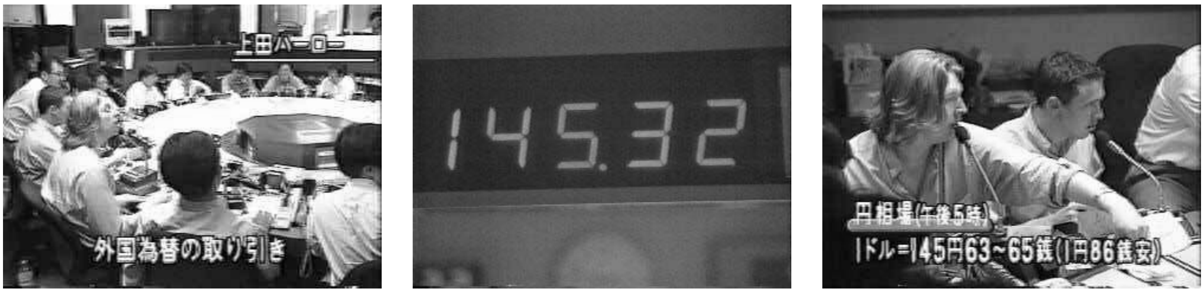


Figure 1: Procedure of discourse structure analysis for news video

Figure 1 shows the procedure of discourse structure analysis for news video. We applied our method to NHK[1] News [2]. This method is aimed to make the process of retrieval, summarization, and information extraction more efficient.

[1] Nippon Hoso Kyokai (Japan Broadcasting Corporation)

[2] NHK news reports do not have closed captions. Instead of closed captions, we used scripts which were read out by newscasters as transcripts.



shot (a) shot (b) shot (c)

(I) The US dollar inched up against the yen as the stock market continued the selling trend in Tokyo today.

(II) The US currency traded at 145.63–65 yen at 5 p.m. Tokyo.

Figure 2: An example of shots and their transcript in a news video (NHK evening news, August/3/1998)

2. DISCOURSE STRUCTURE AND VIDEO

Little attention has been given to discourse structure of video data in image processing. This is because it is difficult to determine it only by analyzing image data. In contrast to this, discourse structure is the subject of a large number of studies in natural language processing. So several methods for estimating the discourse structure of a text have been explored[Sumita 92] [Kurohashi 94]. Therefore, these methods can be applied to language data of video data in order to determine discourse structure in video data.

In addition, some researchers in natural language processing showed that the information of discourse structure is useful for extracting significant sentences and summarizing a text [Miike 94] [Marcu 97]. It suggests that information of video discourse structure is utilized for extracting significant video segments and skimming. It may be useful to look at video skimming and extraction of significant segments before we discuss some points about discourse structure analysis because they are closely related to the discourse structure estimation.

One of the simple ways to skim a video is by using the pair of the first frame/image of the first shot and the first sentence in the transcript. However, this representative pair of image and language is often a poor topic explanation. To solve this problem, Zhang, et.al, proposed a method for key-frame selection by using several image features such as colors, textures, and temporal features including camera operations [Zhang 95]. Also, Smith and Kanade proposed video skimming by selecting video segments based on TFIDF, camera motion, human face, captions on video, and so on [Smith 97]. These techniques are broadly applicable, however, still have problems. One is the semantic classification of each segment. To solve this problem, Nakamura and Kanade proposed the spotting by association method which detects relevant video segments by associating image data and language data [Nakamura 97]. Also, Watanabe, et.al, proposed a method for analyzing telops (captions) in video news reports by using layout and language information [Watanabe 96]. However, these studies did not deal with coherence relations between video segments.

In contrast to this, several works on discourse structure have been made by researchers in natural language processing. Pursuing these studies, we are confronted with two points of discourse structure analysis:

- available knowledge for estimating discourse structure, and
- definition for discourse units and coherence relations.

First, we shall discuss the available knowledge for estimating discourse structure. Most studies on discourse structure have focused on such questions as what kind of knowledge should be employed, and how inference may be performed based on such knowledge (e.g., [Grosz 86], [Hobbs 85], [Zadrozny 91]). In contrast to this, Kurohashi and Nagao pointed out that a detailed knowledge base with broad coverage is unlikely to be constructed in the near future, and that we should analyze discourses using presently available knowledge. For these reasons, they proposed a method for estimating discourse structure by using surface information in sentences [Kurohashi 94]. In video analysis, the same problems occurred. Therefore, we propose here a method for estimating the discourse structure in a news report by using surface information in the transcript.

Next, we shall discuss the definition for discourse unit and coherence relation. As mentioned, discourses are composed of segments (discourse units), and these are connected to previous ones by coherence relations. However, there has been a variety of definitions for discourse unit and coherence relation. For example, a discourse unit can be a frame, a shot, or a group of several consecutive shots. In this study, we consider as a discourse unit, one or more shots which are associated with one part of announcer's speech. For example, shot (a), (b), and (c) in Figure 2 represent consecutive shots in the news video "Both yen and stock were dropped"(Aug/3/1998), while sentence (I) and (II) are parts of the transcript. Sentence (I) was spoken in shot (a) and (b), and correspondingly, sentence (II) was spoken in shot (c). As a result, shot (a) and (b) were merged and the result was considered as one discourse unit. On the other hand, shot (c) alone constituted one discourse unit. We will explain how to extract the discourse units in Section 3.1.

In contrast to this, coherence relations strongly depend on the genre of video data: dramas, documentaries, news reports, sports castings, and so on. From the number of coherence relations suggested so far, we selected the following relations for our target, news reports:

List: S_i and S_j involve the same or similar events or states, or the same or similar important constituents

Contrast: S_i and S_j have distinct events or states, or contrasting important constituents

Topic chaining: S_i and S_j have distinct predications about the same topic

Topic-dominant chaining: A dominant constituent apart from a given topic in S_i becomes a topic in S_j

Elaboration: S_j gives details about a constituent introduced in S_i

Reason: S_j is the reason for S_i

Cause: S_j occurs as a result of S_i

Example: S_j is an example of S_i

where S_i denotes the former segment and S_j the latter.

3. ESTIMATION OF DISCOURSE STRUCTURE

Our determination of how video segments are combined and what kind of coherence relations are involved is made in the next way:

1. extract discourse units from a news report,
2. extract three kinds of clue information from transcripts, and then, transform them into reliable scores for some relations, and
3. choose the connected sentence and the relation having the maximum reliable score. If two or more connected sentences have the same maximum score, the chronological nearest segment is selected.

3.1. Extraction of Discourse Units

A shot is generally regarded as a basic unit in video analysis. In this study, however, not only a shot but also more consecutive ones are considered a basic unit (discourse unit). This is because there are some cases where several consecutive shots correspond with one sentences in a transcript. In this case, these consecutive shots should be regarded as a discourse unit. In contrast to this, one shot should be regarded as a discourse unit when it correspond with one or more sentences in a transcript. In both cases, the start/end point of a discourse unit often lies in the pause because the announcer needs to take breath at the end of a sentence. As a result, discourse units are extracted in the next way:

1. detect scene cuts in a video by using DCT components [Iwanari 94],
2. detect speech pauses in the video, and
3. extract the start/end points of discourse units by detecting the cuts in the pause.

For evaluating this method, we used 105 news reports of NHK News. The recall and precision of discourse unit detection were 71% and 97%, respectively, while those of scene change detection were 80% and 90%. We modified the extracted discourse units by hand and used them in the discourse structure analysis described in Section 3.2. In addition, each discourse unit was associated with the corresponding sentences in a transcript by hands. This is because NHK news reports do not have closed captions.

3.2. Detection of Coherence Relations

In order to extract discourse structure, we use three kinds of clue information in transcripts:

- clue expressions indicating some relations,
- occurrence of identical/synonymous words/phrases in topic chaining or topic-dominant chaining relation, and
- similarity between two sentences in list or contrast relation.

Then they are transformed into reliable scores for some relations. In other words, as a new sentence (NS) comes in, reliable scores for all possible connected sentences and relations are calculated by using above three types of clues. As a final result, we choose the connected sentence (CS) and the relation having the maximum reliable score.

3.2.1. Detection of Clue Expressions

In this study, we use 41 heuristic rules for finding clue expressions by pattern matching and relating them to proper relations with reliable scores. A rule consists of two parts: (1) conditions for rule application and (2) corresponding relation and reliable score. Conditions for rule application consist of four parts:

- rule applicable range,
- relation of CS to its previous DS,
- dependency structure pattern for CS, and
- dependency structure pattern for NS.

Pattern for CS and NS are matched not for word sequences but for dependency structures of both sentences. We apply each rule for the pairs of a CS and NS. If the condition of the rule is satisfied, the specified reliable score is given to the corresponding relation between the CS and the NS.

For example, Rule-1 in Figure 3 gives a score (20 points) to the reason relation between two adjoining sentences if the NS starts with the expression "*nazenara* (because)". Rule-2 in Figure 3 is applied not only for the neighboring CS but also for farther CSs, by specifying the occurrence of identical words "X" in the condition.

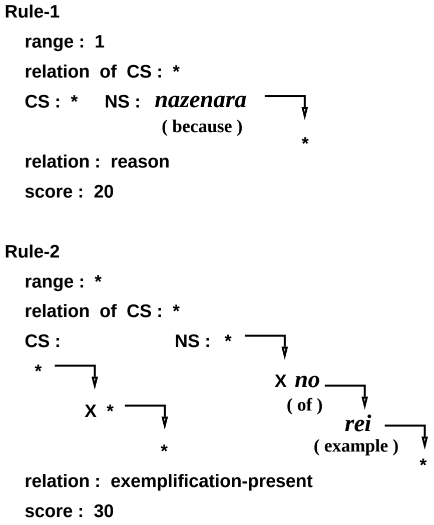


Figure 3: Examples of heuristic rules for clue expressions

3.2.2. Detection of Word/Phase Chain

In general, a sentence can be divided into two parts: a topic part and a non-topic part. When two sentences are in a topic chaining relation, the same topic is maintained through them. Therefore, the occurrence of identical/synonymous word/phrase (the word/phrase chain) in topic parts of two sentences supports this relation. On the other hand, in the case of topic-dominant chaining relation, a dominant constituent introduced in a non-topic part of a prior sentence becomes a topic in a succeeding sentence. As shown, the word/phrase chain from a non-topic part of a prior sentence to a topic part of a succeeding sentence supports this relation.

For these reasons, we detect word/phrase chains and calculate reliable scores in the next way:

1. give scores to words/phrases in topic and non-topic parts according to the degree of their importance in sentences,
2. give scores to the matching of identical/synonymous words/phrases according to the degree of their agreement, and
3. give these relations the sum of the scores of two chained words/phrases and the score of their matching.

For example, by Rule-a and Rule-b in Figure 4, words in a phrase whose head word is followed by a topic marking postposition "*wa*" are given some scores as topic parts. Also, a word in a non-topic part in the sentential style, "*ga aru* (there is ...)" is given a large score (11 points) by Rule-c in Figure 4 because this word is an important new information in this sentence and topic-dominant chaining relation involving it often occur. Matching of phrases

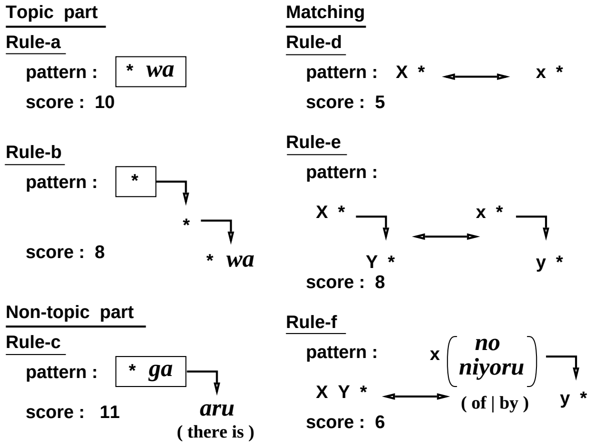


Figure 4: Examples of rules for topic/non-topic parts

like "A of B" is given a larger score (8 points) by Rule-e than that of word like "A" alone by Rule-d (5 points) in Figure 4.

3.2.3. Calculation of Similarity between Sentences in a Transcript

When two sentences have list or contrast relation, they have a certain similarity. As a result, we measure such a similarity for finding list or contrast relation in the next way. First, the similarity value between two words are calculated according to exact matching, matching of their parts of speech, and their closeness in a thesaurus dictionary. Second, the similarity value between two word-strings is calculated roughly by combining the similarity values between words in the two word-strings with the dynamic programming method for analyzing conjunctive structures [Kurohashi 94]. Then, we give the normalized similarity score between a CS and an NS to their list and contrast relations as a reliable score.

4. EXPERIMENTS AND DISCUSSION

For evaluating this method, we used 22 news reports of NHK News. Each report was a few minutes in length. The experimental results are shown in Table 1. As mentioned, news reports of NHK News do not have closed captions. For this reason, each video segment (discourse unit) was associated with the corresponding sentences in a transcript by hands.

Table 1: Analysis results

Relation	Success	Failure
List	2	1
Contrast	1	0
Topic chaining	38	11
Topic-dominant chaining	20	2
Elaboration	0	3
Reason	0	0
Cause	4	0
Example	0	0
Total	65	17

Figure 5 shows the video news report we used in our experiment. As shown, shot (b) and (c) were merged together because there was no pause at the cut point between them. Sentence (I), (II), (III), (IV), and (V) were associated with shot (a), (b)(c), (d), (e), and (f), respectively. Coherence relations between video segments were estimated in the following way: a topic-dominant chaining relation was estimated between shot (a) and (b)(c) because "Prime Minister Obuchi" was found in the topic part of sentence (II) and in the non-topic part of sentence (I). The same relation was also estimated between shot (b)(c) and (d) because "the Fiscal Structural Reform Law" was found in the topic part of sentence (III) and in the non-topic part of sentence (II). On the contrary, topic chaining relation was estimated between shot (a) and (e) because "the Ministry of Finance" was found in the topic parts of sentence (I) and (IV). In this case, the system detected also another relation: a topic-dominant chaining relation between shot (b) and (e). However, the system selected the former one because the former exceeded the latter in score. The system also determined a topic chaining relation between shot (e) and (f). In this case, the system additionally detected two other relations: topic chaining relation between shot (a) and (f), and topic-dominant chaining relation between shot (b) and (f). But the relation between (a) and (f) was chosen, because its reliable score was greater than the score between (b) and (f) and equal to the score between (e) and (f), but there the distance between the shots was greater. Figure 6 shows the result of this analysis.

As shown in Table 1, 11 topic chaining and 2 topic-dominant chaining relations could not be extracted. The reasons were (1) the topic words of the following sentences were omitted [3] and (2)

[3] There are many ellipses in Japanese sentences.

	shot	transcript
(a)		(I) In accordance with instructions of Prime Minister Obuchi, the Ministry of Finance will decide about new guidelines for budget requests which is free from the restrictions of the Fiscal Structural Reform Law.
(b)		(II) Prime Minister Obuchi called Vice Minister Tanami, the Ministry of Finance, into the Official Residence, and instructed him to make new budget request guidelines in line with the freeze policy of the Fiscal Structural Reform Law.
(c)		
(d)		(III) The Fiscal Structural Reform Law sets upper limits for expenditures in all categories but social security.
(e)	大蔵省 公共事業・中小企業関係費 ⇒上限超え認める方向で新基準策定へ	(IV) In accordance with prime minister's instruction, the Ministry of Finance establishes new guidelines in which key government expenditures, for example, public works projects, are permitted to go beyond the limits of the Law.
(f)		(V) the Ministry of Finance will decide about the new guidelines by the middle of the next week, and government ministries and agencies will submit their budget requests in accordance with this guideline by the end of the month.

Figure 5: An example of news video ("New guidelines for budget requests", NHK evening news, August/3/1998)

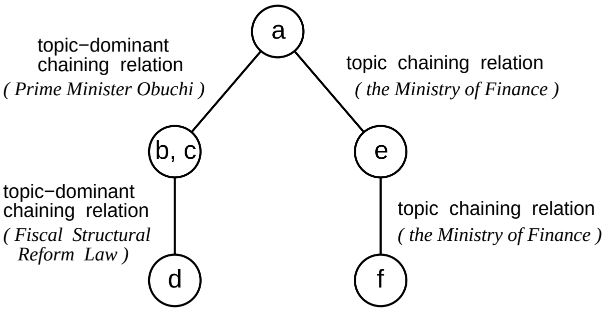

a
topic−dominant chaining relation
(Prime Minister Obuchi)
topic chaining relation
(the Ministry of Finance)
b, c
e
topic−dominant chaining relation
(Fiscal Structural Reform Law)
topic chaining relation
(the Ministry of Finance)
d
f

Figure 6: The result of discourse structural analysis for the news video shown in Figure 5

the topic word was changed (e.g., driver → man who drove the car) or abbreviated. Also, 3 elaboration relations could not be extracted. This was because there were no clue expressions for the elaboration relation in the sentences. However, the system could mostly detect clue expressions and occurrence of identical/synonymous words/ phrases. In some cases (e.g., a compound sentence), there were many clues for an NS supporting various relations to several CSs. The system could detect them, however, extracted only one CS and relation. In this study, we introduce a reliable score for determining the most plausible CS and relation. As shown in Table 1, this method is useful, however, we should investigate a method for extracting more CSs and relations than one when several CSs and relations exist.

In this study, we assumed that image and language data correspond to the same portion of a news report. For this reason, it is likely that the relation between images slightly differs from the analysis result when image and language are taken form different portions (correspondence problem between image and language).

At the end of this section, we discuss video summarization using discourse structure information. First, we consider summarization of the news video shown in Figure 5 with the summarization topic concerning the Ministry of Finance. The summarization system traces topic chaining relations and generates video summarization which consists of shots (a), (e), and (f). Next, we consider summarization of the same news video with the summarization topic concerning the Prime Minister. The system traces a topic-dominant chaining relation and generates video summarization which consists of shot (a), (b), and (c).

References

[Grosz 86] Grosz and Sidner: Attention, Intentions, and the Structures of Discourse, Computational Linguistics, 12-3, (1986).

[Hobbs 85] Hobbs: On the Coherence and Structure of Discourse, Technical Report No. CSLI-85-37, (1985).

[Iwanari 94] Iwanari and Ariki: Scene Clustering and Cut Detection in Moving Images by DCT components, (in Japanese), technical report of IEICE, PRU-93-119, (1994).

[Kurohashi 92] Kurohashi and Nagao: Dynamic Programming Method for Analyzing Conjunctive Structures in Japanese, COLING-92, (1992).

[Kurohashi 94] Kurohashi and Nagao: Automatic Detection of Discourse Structure by Checking Surface Information in Sentences, COLING-94, (1994).

[Marcu 97] Marcu: From Discourse Structures to Text Summaries, ACL workshop on Intelligent Scalable Text Summarization, (1997).

[Miike 94] Miike, Itoh, Ono, and Sumita: A Full-Text Retrieval System with a Dynamic Abstract Generation Function, SIGIR-94, (1994).

[Nagao 00] Nagao, Shirai, and Hashida: Multimedia Data Summarization Based on the Global Document Annotation, (in Japanese), 6th Annual Meeting of The Association for Natural Language Processing, (2000).

[Nakamura 97] Nakamura and Kanade: Semantic Analysis for Video Contents Extraction – Spotting by Association in News Video, ACM Multimedia 97, (1997).

[Smith 97] Smith and Kanade: Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques, IEEE CVPR, (1997).

[Sumita 92] Sumita, Ono, Chino, Ukita, and Amano: A Discourse Structure Analyzer for Japanese Text, International Conference of Fifth Generation Computer Systems, (1992).

[Watanabe 96] Watanabe, Okada, and Nagao: Semantic Analysis of Telops in TV Newscasts, (in Japanese), technical report of Information Processing Society of Japan, NL-116–16, (1996).

[Zadrozny 91] Zadrozny and Jensen: Semantics of Paragraphs, Computational Linguistics, 17-2, (1991).

[Zhang 95] Zhang, Low, Smoliar, and Wu: Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution, ACM Multimedia 95, (1995).

SECTION 3

Semantic Annotation of Document Segments

Alignment of Sound Track with Text in a TV Drama

Seigo Tanimura, Hiroshi Nakagawa
Information Technology Center, The University of Tokyo.
7-3-1 Hongo, Bunkyo, Tokyo, JAPAN 113-0033
{tanimura,nakagawa}@r.dl.itc.u-tokyo.ac.jp

Abstract

We propose a system to align a sound track and a part of TV drama video contents with a script. We first use the number of moras in each sentence of speech line, the sounding time in a sound track and the shot change time in the motion image to align them approximately. Then we perform DP matching to align a sequence of words obtained from a speech recognition system applied to a sound track with each sentence of speech lines in a script. Confident correspondences obtained from the DP matching of the words act as the pivots to improve alignment accuracy iteratively. The results show that around a half of the sentences in a script were aligned within the differences of up to two seconds.

1 Introduction

Alignment of video to text is essential to make video contents flexibly reusable. Although it seems to be promising for this purpose to apply a speech recognition technology to the sound track of a video content, speech recognition for a TV drama or a feature film is actually difficult. The major obstacle significant in a drama and film is the poor accuracy of speech recognition. Due to the background music, the noise from the environment in a location and audio compression like MPEG, the accuracy of speech recognition is only around 10-30%. Thus the result of speech recognition to the sound track of a drama or film does not have enough quality to directly reuse the video contents.

Although the result of speech recognition is of no use by itself, we still have the script of a drama or film. Hence alternative method is to align the script of a drama or film to the video. Several proporsals have been made to solve this problem. Yaginuma et al. (Yaginuma and Sakauchi, 1996) proposed time alignment of a TV drama by the physical shot changes in video, the volume in the sound track and the number of characters in the speech lines of the script. However, the accuracy of alignment by their method achieved only 70% for a sentence due to lack of speech recognition in their method.

While speech recognition can improve the accuracy of alignment, we need to solve a couple of problems to apply speech recognition to a sound track of a drama or film. A state-of-the-art speech recognition system performs speech recognition in a sentence-to-sentence manner. In addition to that, applying speech recognition directly to a whole length of sound track, say 30 minutes or longer, involves unrealistic costs in both time and memory. Hence a sound track needs to be divided into sentences prior to applying a speech recognition system. Due to the large number of sentences in a TV drama, it may result in inferior accuracy to divide a whole sound track into sentences. We can avoid this by dividing a sound track roughly first into, say, logical scenes. Then we apply more precise segmentation based on the sentences to the logical scenes.

We developed a system to align each sentence of speech lines in a script written in Japanese to the corresponding part of a sound track accompanying the script spoken in Japanese. Our proposing system consists of six modules. Figure 1 shows the architecture of our system. Note that "x alignment" means "alignment of sequences formed from x" in the rest of this paper.

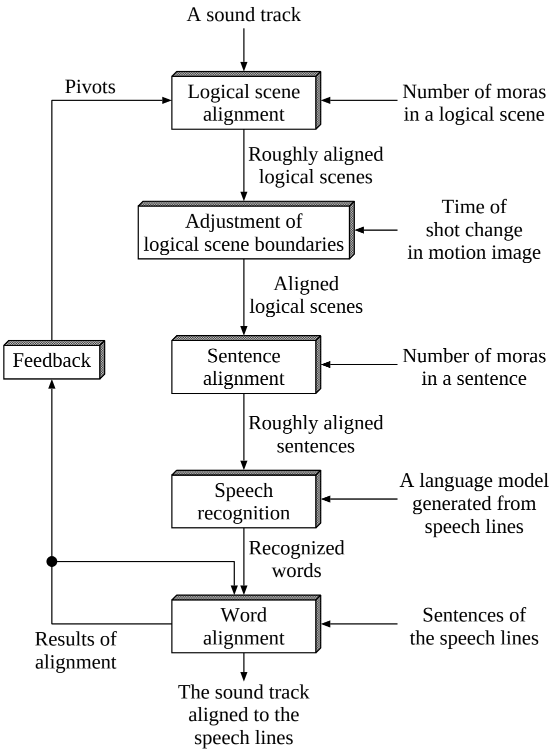

A sound track
Pivots
Logical scene alignment
Number of moras in a logical scene
Roughly aligned logical scenes
Adjustment of logical scene boundaries
Time of shot change in motion image
Aligned logical scenes
Feedback
Sentence alignment
Number of moras in a sentence
Roughly aligned sentences
Speech recognition
A language model generated from speech lines
Recognized words
Word alignment
Sentences of the speech lines
Results of alignment
The sound track aligned to the speech lines

Figure 1: The architecture of our system

In section 2, we describe logical scene alignment and logical scene boundary adjustment. The sentence alignment module is discussed in section 3. We describe the speech recognition module in 4. Section 5 details the word alignment module. The mechanism of feedback is discussed in section 6. We present our experimental results in section 7 and conclude in section 8.

2 Logical Scene Alignment

The word alignment module in our system is capable of aligning a scene of only up to around several thousand words, or several minutes. A scene longer than, say, 10 minutes cannot be aligned directly by word-to-word manner with a practical computational cost. Hence we need to develop other means of approximate alignment capable of handling a scene of 10-30 minutes.

A logical scene in a script can be an alternate of a word to align a sound track to a script. A logical scene has the following features:

- A logical scene can be extracted from a script as easily as a word. A typical logical scene consists of the scene number and the title, followed by the speech lines and directions.
- A logical scene in a motion image often begins and ends at a shot change. It can be detected by the shot detection system which usually uses the change of color histograms between a couple of consecutive frames in a motion image.

These facts imply that a logical scene can be extracted both from a script and motion image. Hence we adopt a logical scene as an alternative of a word to align a sound track to the script. The problem is that a logical scene may consist of not a single but several shots. Thus duration of a logical scene needs to be estimated using the speech lines of a script.

It is well known in Japanese linguistics and phonetics that a length of utterance duration in Japanese is basically proportional to the number of the moras in the uttered word (Kubozono, 1999). Counting this fact, we seek the starting and ending time of a part of the sound track which corresponds to a logical scene in the script as the first approximation, so that the duration of sounding segment corresponding to every logical scene of the script gets proportional to the number of moras in the logical scene. The starting and ending time are then adjusted to the nearest shot change time in the motion image. The sounding segments in the sound track are extracted by comparing power of every 25msec period in the sound track to the predetermined threshold. The number of moras in a logical scene is computed by the pronunciation of the words in the logical scene obtained from a morphological analyzer. We apply a commercial scene cut tool (Hitachi, Ltd., 1997) to detect the shot changes in the motion image.

3 Sentence Alignment

A logical scene generally consists of several sentences. However, unlike extracting the logical scenes in a whole drama, it is difficult to extract the sentences directly from a sound track or a sequence of image frames. To segment out each sentence in the sound track, we apply the method to use the numbers of moras and the duration of sounding segments described in section 2, except that we seek a part of the sound track corresponding to not a logical scene but a sentence. Since an utterance of a speech line may not begin or end at a shot change, we modify the alignment method proposed in section 2 by omitting adjustment of starting and ending time to the nearest shot change time.

4 Speech Recognition

4.1 Language and Acoustic Models

In our system, the words to be uttered are given as the speech lines of a script in advance. A word bigram language model used during speech recognition is generated from the whole speech lines to improve the accuracy of a speech recognition system. Every word in a language model is uttered at least once, and no other words are uttered with this tailored language model. This avoids recognizing words not appearing in the speech lines.

It also brings about a better accuracy in speech recognition to choose the acoustic model adapted for a speaker. However, a general method of speaker adaptation has some major problems in our system, that is, the speakers in the segments of the sound track are not given as the input. Furthermore, not only one but several speakers may utter in a single segment of the sound track. Thus we cannot determine a suitable acoustic model prior to speech recognition. In order to solve this problem, we perform speech recognition with multiple acoustic models (Ming et al., 1999), say female and male models, because the difference of these two models affects the accuracy of a speech recognition system significantly. To perform speech recognition simply, these acoustic models are used in parallel. This allows us to improve the accuracy of sentence alignment without misselecting a suitable acoustic model, described in section 5.2.

4.2 A Speech Recognition System and Filtering out Noisy Words

We use JULIUS (Ito et al., 1998) as a speech recognition system. A language model is generated by applying a Japanese morphological analyser JUMAN (Kurohashi and Nagao, 1997) and CMU-Cambridge SLM Toolkit (Clarkson, 1997) to the speech lines. We use HMMs of 16 mixed density for triphones of 3000 states as the acoustic models. HMMs for female and male speakers are used in parallel.

We postprocess the recognized words in order to improve the accuracy of sentence alignment. A speech recognition system treats unuttering duration in a sound track as a comma or a full stop. However, these do not usually match to the commas and full stops in the speech lines. Thus commas and full stops in the recognized words and speech lines are apparently noise in word alignment. We filter out these noisy words from the recognized words and the words in the speech lines prior to word alignment.

5 Word Alignment with DP Matching

In this module, we align for each of logical scenes the sequences of the recognized words to the sentences of the speech lines based on the similarity between a pair of words. The similarity between words involved in this alignment is computed by performing mora-based DP matching. More precisely, our word alignment system consists of two level alignment modules; a word alignment module for a pair of sentences described in section 5.2 and a mora alignment for a pair of words described in section 5.1. To proceed the word alignment for sentences, the word alignment module invokes the mora alignment module for words interactively.

5.1 Mora Alignment for Words

We describe in this section our mora-based DP matching. Let A_m be a sequence of moras of a word in a sentence of a speech line in the script, and B_m be a recognized word of the speech line, respectively. They are defined as follows(A subscript of m stands for a mora):

$$\begin{aligned} A_m &= \{a_{m1}, a_{m2}, \ldots, a_{mi}, \ldots, a_{mI}\} \quad (1) \\ B_m &= \{b_{m1}, b_{m2}, \ldots, b_{mj}, \ldots, b_{mJ}\} \quad (2) \end{aligned}$$

where a_{mi} is the ith mora in a word of a sentence, b_{mj} is the jth mora in a recognized word, I is the number of moras in the word of a sentence, and J is the number of the moras in a recognized word. Then we define a similarity between a pair of moras, $s_m(a_{mi}, b_{mj})$ as follows:

$$s_m(a_{mi}, b_{mj}) = \begin{cases} 3 & (a_{mi} = b_{mj}) \\ 2 & \left(\begin{array}{l} \text{Only the vowel of } a_{mi} \text{ is} \\ \text{equal to the vowel of } b_{mj} \end{array}\right) \\ 0 & (\text{None of the above}) \end{cases} \quad (3)$$

A vowel in a recognized word is more confident than a consonant in general. Thus we give a similarity to a pair of moras with the identical vowel and different consonants as well. Using the expression (3), we iterate an expression (6) with the initial conditions of expressions (4) and (5) as follows to compute $g_m(a_{mI}, b_{mJ})$:

$$g_m(a_{m1}, b_{m1}) = 0 \quad (4)$$

$$g_m(a_{mi}, b_{m1}) = g_m(a_{m1}, b_{mj}) = -\infty \quad (5)$$

$$g_m(a_{mi}, b_{mj}) = \max_{q=1,2,\ldots,p-1} \begin{cases} g_m(a_{mi-q-1}, b_{mj-1}) + s_m(a_{mi-q}, b_{mj}) \\ g_m(a_{mi-1}, b_{mj-1}) + s_m(a_{mi}, b_{mj}) \\ g_m(a_{mi-1}, b_{mj-q-1}) + s_m(a_{mi}, b_{mj-q}) \\ \quad + \sum_{r=1}^{q} s_m(a_{mi}, b_{mj-q+r}) \end{cases} \quad (6)$$

where p is a parameter to forbid stretching and shrinking p or more moras locally.

Figure 2 shows the local constraint and the weights in the expression (6). The black dots show that the alignment paths can grow from these dots to (i, j). The numbers accompanied by the paths are the weights of them. This constraint assures that the possible maximum of $g_m(a_{mI_m}, b_{mJ_m})$ does not depend on J_m. We iterate the expression (6) until $g_m(a_{mI_m}, b_{mJ_m})$ gets computed. Finally, the similarity between A_m and B_m is defined as follows:

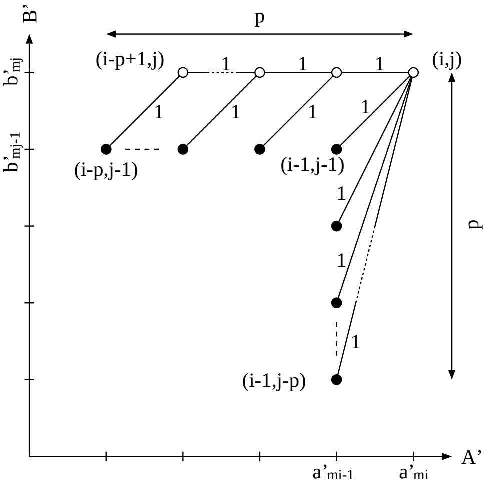


Figure 2: The local constraint and the weights of the paths in mora alignment

$$S_m(A_m, B_m) = \begin{cases} g_m(a_{mI}, b_{mJ}) + \frac{10}{|I-J|+1} & (A_m \neq B_m) \\ \left(g_m(a_{mI}, b_{mJ}) + \frac{10}{|I-J|+1}\right)^2 & (A_m = B_m) \end{cases} \quad (7)$$

The second term in the expression (7) is included to discourage matching words with excess difference between the numbers of the moras. If $A_m = B_m$, i.e. they have the identical pronunciation with each other, we square $g_m(a_{mI_m}, b_{mJ_m}) + \frac{10}{|I_m - J_m|+1}$ to give higher similarity to these words corresponding to each other. We use $S_m(A_m, B_m)$ in word alignment for sentences, described in section 5.2.

5.2 Word Alignment for Sentences

Word alignment algorithm for sentences is also DP matching as well as the mora alignment for words is. Let C_w and D_w be a sequence of words in a sentence and a recognized word of the speech line, respectively. They are defined as follows(A subscript of w

stands for a word):

$$\begin{aligned} C_w &= \{c_{w1}, c_{w2}, \ldots, c_{wi}, \ldots, c_{wI}\} \quad (8) \\ D_w &= \{d_{w1}, d_{w2}, \ldots, d_{wj}, \ldots, d_{wJ}\} \quad (9) \end{aligned}$$

where c_{wi} is the ith word in the whole sentence of the speech line, d_{wj} is the jth word in the recognized words, I is the number of words in the whole sentences, and J is the number of the recognized words. Then we define a similarity between a pair of words, $s_w\left(c_{wi}, d_{wj}\right)$ using the results of the mora alignment module for words described in section 5.1 as follows:

$$s_w\left(c_{wi}, d_{wj}\right) = S_m\left(A_{mi}, B_{mj}\right) \qquad (10)$$

where A_{mi} and B_{mj} are the sequences of moras in the words c_{wi} and d_{wj} respectively, and $S_m\left(A_{mi}, B_{mj}\right)$ is defined in the expression (7). Using the expression (10), we iterate an expression (13) with the initial conditions of expressions (11) and (12) as follows to compute $g_w\left(c_{wI}, d_{wJ}\right)$:

$$g_w\left(c_{w1}, d_{w1}\right) = 0 \qquad (11)$$

$$g_w\left(c_{wi}, d_{w1}\right) = g\left(c_{w1}, d_{wj}\right) = -\infty \qquad (12)$$

$$g_w\left(c_{wi}, d_{wj}\right) = \max_{q=1,2,\ldots,p-1} \begin{cases} g_w\left(c_{wi-q-1}, d_{wj-1}\right) + 2s_w\left(c_{wi-q}, d_{wj}\right) \\ \quad + \sum_{r=1}^{q} s_w\left(c_{si-q+r}, d_{wj}\right) \\ g_w\left(c_{wi-1}, d_{wj-1}\right) + 2s_w\left(c_{wi}, d_{wj}\right) \\ g_w\left(c_{wi-1}, d_{wj-q-1}\right) + 2s_w\left(c_{wi}, d_{wj-q}\right) \\ \quad + \sum_{r=1}^{q} s_w\left(c_{wi}, d_{wj-q+r}\right) \end{cases} \qquad (13)$$

where p is a parameter to forbid stretching and shrinking p or more words locally.

Figure 3 shows the local constraint and the weight of the paths. The black dots show that the alignment paths can grow from these dots to (i, j). The numbers accompanied by the paths are the weights of them. The expressions (11) and (12) ensure that the first words in the sequences always correspond to each other. We iterate the expression (13) until $g_w\left(c_{wI}, d_{wJ}\right)$ gets computed.

5.3 Selection of the actually uttered words

As mentioned in section 4, we obtain two sequences of recognized words with female and male acoustic models respectively. In order to select the actually uttered words dynamically between these two sequences, we developed a method to select the appropriate sequence upon finding a word d_{i-1} uttered prior to d_i.

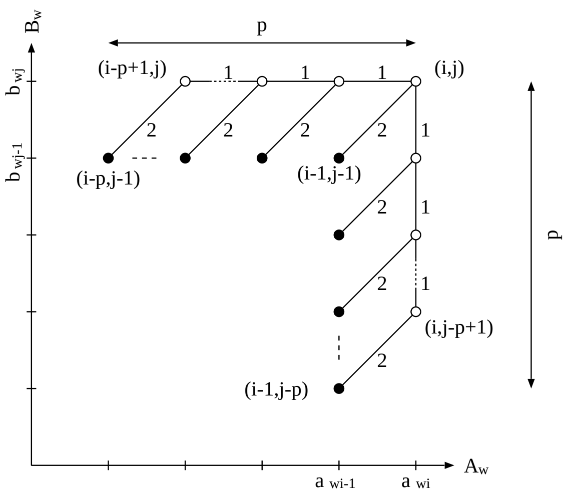


Figure 3: The local constraint and the weights of the paths in DP matching of the word sequences

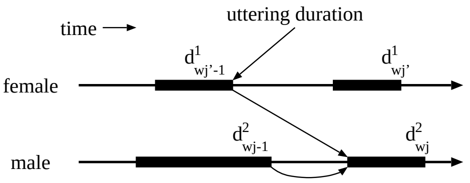


Figure 4: Selection of words uttered prior to d^2_{wj}

Figure 4 shows our method to select the words uttered prior to d^2_{wj} from the words recognized by acoustic models for female and male. We select from each of the word sequences the words with the nearest ending time to the beginning time of d^2_{wj} from all of the words with the ending times prior to the beginning time of d^2_{wj}. For example, we find $d^1_{wj'-1}$ and d^2_{wj-1} as the words uttered prior to d^2_{wj} in figure 4. Then each of $d^1_{wj'-1}$ and d^2_{wj-1} is substituted into d_{wj-1} to compute $g_w\left(c_{wi}, d^2_{wj}\right)$. Then the word that gives the highest $g_w\left(c_{wi}, d_{wj}^2\right)$ is selected as d_{wj-1}.

6 Improvement by Feedback

We improve the accuracy of alignment by fixing confident correspondences in the result of the word alignment module discussed in section 5 as fixed pivots in logical scene alignment.

A confident correspondence should not consist of short words. Short words in our system refer to the words of only one or two moras. Most of such the words are functional words in Japanese. They appear quite frequently in any speech lines. Moreover, a speech recognition system may misrecognize utterances or even noises to end up with spurious functional words. On the contrary, a correspondence of long words with an identical pronunciation can be confident. Such a correspondence shows that the utterance is recognized correctly with a matching word in the speech lines. Counting these facts, we define a confident corresponding word pair as follows:

- The corresponding words have an identical pronunciation.
- The pronunciation should have a length of at least three moras.

We pick up the correspondences satisfying both of these conditions shown above from the result of word alignment as the confident correspondences. Using these confident correspondences as fixed pivots, we realign the sounding segments of sound track with each logical scenes and sentences of the speech lines, and reperform the speech recognition and word alignment described in section 2-5, respectively.

7 Experimental Results

We evaluated the alignment accuracy of our system experimentally. Table 1 shows the sample scene for our experiment.

We first counted for each cycle of iteration the number of the recognized words with three or more moras and the number of the aligned sentences including at least one pivot. The results are shown in figure 2 and 3, respectively.

Table 1: Sample scene

Number of logical scenes	24
Number of sentences	91
Number of words with three or more moras	502
Duration of the sound track [min:sec]	14:44

Table 2: The numbers of the recognized words with three or more moras

Iteration	No. of words
Cycle 1	59
Cycle 2	57
Cycle 3	59

Table 3: The numbers of the aligned sentences with at least one pivot

Iteration	No. of sentences
Cycle 1	31
Cycle 2	37
Cycle 3	36

Although the numbers of recognized words shown in table 2 do not cover all of the words in the script, we can still approximate the accuracy of speech recognition by these results. The accuracy of speech recognition stayed around 12%, indicating a poor quality of the sample sound track. Nevertheless a third of the sentences were aligned with pivots. In addition, we gained quite a few number of newly aligned sentences as iteration proceeds, as shown in table 3. These results imply that the pivots in a sentence obtained in the first cycle diffuse to the neighbour sentences.

In order to investigate the effect of the pivot gain shown in table 3, we evaluated the accuracy of alignment by the following method. We measured the difference between the utterance beginning/ending time of the recognized word aligned to the first/last word of each sentence and the correct utterance beginning/ending time of the sentence, which is expressed as ϵ_b/ϵ_e henceforth. We then counted the number of the sentences satisfying $|\epsilon| \leq E$ where ϵ is either ϵ_b or ϵ_e and E is one of 1, 3 or 5 seconds. The average of $|\epsilon|$ for the whole sentences, Av. was also computed. The results are shown in table 4.

Table 4: The numbers of the sentences satisfying $|\epsilon| \leq E$ and the average of $|\epsilon|$

Iteration	ϵ	E			Av.[s]
		1[s]	3[s]	5[s]	
Cycle 1	ϵ_b	17	31	42	16.9
	ϵ_e	7	21	35	18.3
Cycle 2	ϵ_b	32	47	60	8.4
	ϵ_e	16	36	53	10.1
Cycle 3	ϵ_b	31	55	62	9.9
	ϵ_e	19	43	53	12.0

We can state from these results that our alignment system can align not only a sentence recognized correctly but also its neighbor sentences. On the other hand, the average of $|\epsilon|$ did not increase in cycle 3 because the level of noise was extremely higher than the level of utterance for 21 sentences uttered in a running train. Due to the poor accuracy of speech recognition of these 21 sentences, we obtained only 5 pivots at most from these sentences, ending up with the large Av.s shown in table 4.

8 Conclusion

We proposed a system to align a sound track and a sequence of image frames with sentences of speech lines in a TV drama. Our next target is to improve the accuracy of speech recognition and to seek a promising application area of our alignment method.

Acknowledgement

This work is supported in part by the Grant-in-Aid for Creative Basic Research: Multi Media Mediation Mechanism(09NP1401), of the Ministry of Education, Science, Sports and Culture, Japan.

References

Philip Clarkson. 1997. The CMU-Cambridge statistical language modeling toolkit v2. `http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html`.

Hitachi, Ltd. 1997. Mediachef/CUT for Windows 95.

Katsunobu Ito, Tatsuya Kawahara, Kazuya Takeda, and Kiyohiro Shikano. 1998. Japanese dictation toolkit. *Proc. of the 12th Annual Conference of Japanese Society for Artificial Intelligence.* (In Japanese).

Haruo Kubozono. 1999. *Nihongo no Onsei(Phonetics in Japanese).* Iwanami Shoten Publishers. (In Japanese).

Sadao Kurohashi and Makoto Nagao, 1997. *Japanese Morphological Analysis System JUMAN Ver. 3.4.* (In Japanese).

Ji Ming, Philip Hanna, Darryl Stewart, Marie Ownes, and F. Jack Smith. 1999. Improving speech recognition performance by using multi-model approaches. *ICASSP 99*, 1:161–164.

Y. Yaginuma and M. Sakauchi. 1996. Content-based drama editing based on intermedia synchronization. *Proc. of the IEEE Computer Society, International Conference on Multimedia Computing and Systems '96*, pages 322–329, 6.

Semantic Transcoding: Making the World Wide Web More Understandable and Usable with External Annotations

Katashi Nagao
IBM Research,
Tokyo Research Lab.
nagao@trl.ibm.co.jp

Shingo Hosoya
Keio University
punpun@sfc.wide.ad.jp

Yoshinari Shirai
NTT Communication
Science Laboratories
way@cslab.kecl.ntt.co.jp

Kevin Squire
Univ. of Illinois
at Urbana-Champaign
k-squire@uiuc.edu

Abstract

This paper proposes an easy and simple method for constructing a super-structure on the Web which provides current Web contents with new value and new means of use. The super-structure is based on external annotations to Web documents. We have developed a system for any user to annotate any element of any Web document with additional information. We have also developed a proxy that transcodes requested contents by considering annotations assigned to them. In this paper, we classify annotations into three categories. One is linguistic annotation which helps the transcoder understand the semantic structure of textual elements. The second is commentary annotation which helps the transcoder manipulate non-textual elements such as images and sounds. The third is multimedia annotation, which is a combination of the above two types. All types of annotation are described using XML, and correspondence between annotations and document elements is defined using URLs and XPaths. We call the entire process "semantic transcoding" because we deal with the deep semantic content of documents with annotations. The current semantic transcoding process mainly handles text and video summarization, language translation, and speech synthesis of documents including images.

1 Introduction

The conventional Web structure can be considered as a graph on a plane. In this paper, we propose a method for extending such planar graph to a three-dimensional structure that consisting of multiple planar layers. Such metalevel structure is based on external annotations on documents on the Web.

Figure 1 represents the concept of our approach.

A super-structure on the Web consists of layers of content and metacontent. The first layer corrensponds 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 an external annotation. A famous example of external annotations is external links that can be defined outside of the set of link-connected documents. These external links have been discussed in the XML community but they have not yet been implemented in the current Web architecture (W3C XML, 2000).

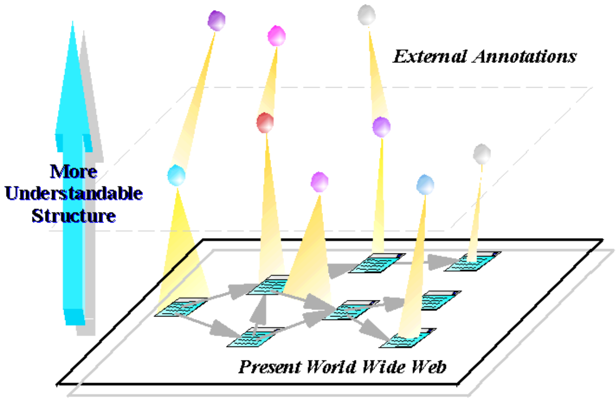


Figure 1: Super-structure on the Web

Another popular example of external annotation is comments or notes on Web documents created by people other than the author. This kind of annotations is helpful for readers evaluating the documents. For example, images without alternative descriptions are not understandable for visually-challenged people. If there are comments on these images, these people will understand the image contents by listening to them via speech transcoding. This example is explained later in more detail.

We can easily imagine that an open platform for creating and sharing annotaions would greatly extend the expressive power and value of the Web.

1.1 Content Adaptation

Annotations do not just increase the expressive power of the Web but also play an important role in content reuse. An example of content reuse is, for example, the transformation of content depending on user preferences.

Content adaptation is a type of transcoding which considers a users' environment such as devices, network bandwidth, profiles, and so on. Such adaptation sometimes also involves a deep understanding of the original document contents. If the transcoder fails to analyse the semantic structure of a document, then the results may cause user misunderstanding.

Our technology assumes that external annotations help machines to understand document contents so that transcoding can have higher quality. We call such transcoding based on annotation "semantic transcoding."

1.2 Knowledge Discovery

Another use of annotations is in knowledge discovery, where huge amounts of Web contents are automatically mined 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 are developing an information collector that gathers documents related to a topic and generates a document containing a summary of each document.

There are many unresolved issues before we can realize true knowledge discovery, but we can say that annotations facilitate this activity.

2 External Annotation

We have developed a simple method to associate 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 have also developed an annotation server that maintains the relationship between contents and annotations and transfers requested annotations to a transcoder.

Our annotations are represented as XML formatted data and divided into three categories: linguistic, commentary, and multimedia annotation. Multimedia (especially video) annotation is a combination of the other two types of annotation.

2.1 Annotation Environment

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

The annotation environment is shown in Figure 2.

The process flows as follows (in this example case, HTML files are processed):

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.

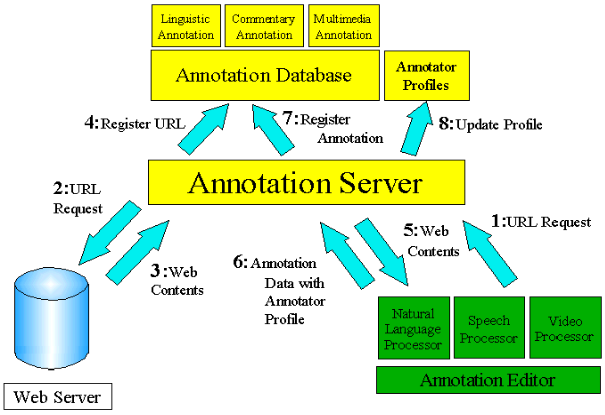


Figure 2: Annotation environment

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 with some personal data (name, professional areas, etc.).
7. The server receives the annotation data and relates it with its URL in the database.
8. The server also updates the annotator profiles.

2.2 Annotation Editor

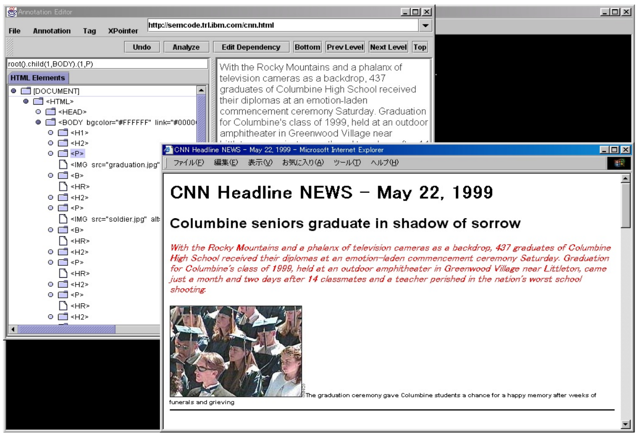
Our annotation editor, implemented as a Java application, can communicate with the annotation server explained below.

The annotation editor has the following functions:

1. To register targets of annotation to the annotation server by sending URLs
2. To specify any element in the document using the Web browser
3. To generate and send annotation data to the annotation server
4. To reuse previously-created annotations when the target contents are updated

An example screen of our annotation editor is shown in Figure 3.

The left window of the editor shows the document object structure of the HTML document. The right window shows some text that was selected on the Web browser (shown on the lower right corner). The selected area is automatically assigned an XPath (i.e., a location identifier in the document) (W3C XPath, 2000).



http://semcode.trl.ibm.com/cnn.html

root().child(1,BODY).(1,P)

```
[DOCUMENT]
  <HTML>
    <HEAD>
    <BODY bgcolor="#FFFFFF" link="#0000
      <H1>
      <H2>
      <P>
        <IMG src="graduation.jpg"
      <B>
        <HR>
      <H2>
      <P>
        <IMG src="soldier.jpg" alt
      <B>
        <HR>
      <H2>
      <P>
        <HR>
      <H2>
      <P>
        <HR>
      <H2>
```

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

CNN Headline NEWS – May 22, 1999

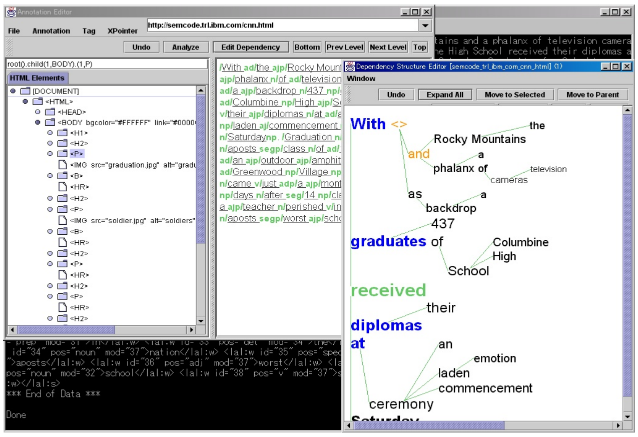
Columbine seniors graduate in shadow of sorrow

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.

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

Figure 3: Annotation editor

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



http://semcode.trl.ibm.com/cnn.html

root().child(1,BODY).(1,P)

```
[DOCUMENT]
  <HTML>
    <HEAD>
    <BODY bgcolor="#FFFFFF" link="#0000
      <H1>
      <H2>
      <P>
        <IMG src="graduation.jpg" alt="gradu
      <B>
        <HR>
      <H2>
      <P>
        <IMG src="soldier.jpg" alt="soldiers"
      <B>
        <HR>
      <H2>
      <P>
        <HR>
      <H2>
      <P>
        <HR>
      <H2>
```

```
/With ad/the ajp/Rocky Mount
ajp/phalanx n/of ad/television
ad/a ajp/backdrop n/437 np/
ad/Columbine np/High ajp/S
v/their ajp/diplomas n/at ad/a
np/laden aj/commencement
n/Saturdaynp. /Graduation n/
n/aposts segp/class n/of ad/
ad/an ajp/outdoor ajp/amphit
ad/Greenwood np/Village np
n/came v/just adp/a ajp/mont
np/days n/after seg/14 np/cla
a ajp/teacher n/perished v/in
n/aposts segp/worst ajp/scho
```

```
tains and a phalanx of television camera
e High School received their diplomas a
```

```
 id="34" pos="noun" mod="37">nation</lal:w> <lal:w id="35" pos="spec
">aposts</lal:w> <lal:w id="36" pos="adj" mod="37">worst</lal:w> <la
pos="noun" mod="32">school</lal:w> <lal:w id="38" pos="v" mod="37">s
:w></lal:s>
*** End of Data ***

Done
```

Figure 4: Annotation editor with linguistic structure editor

2.3 Annotation Server

Our annotation server receives annotation data from any annotator and classifies it according to the annotator. The server retrieves documents from URLs in annotation data and registers the document hash codes with their URLs in its annotation database. The hash codes are used to find differences between annotated documents and updated documents identified by the same URL. A hash code of document internal structure or DOM (Document Object Model) enables the server to discover modified elements in the annotated document (Maruyama et al., 1999).

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 (described below), 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 created by the specified annotator.

We are currently developing a mechanism for access control between annotation servers and normal Web servers. If authors of original documents do not want to allow anyone to annotate their documents, they can add a statement about it in the documents, and annotation servers will not retrieve such contents for the annotation editors.

2.4 Linguistic Annotation

The purpose of linguistic annotation is 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 is currently available. The new tag set was proposed by the GDA (Global Document Annotation) project (GDA, 2000). It is based on XML (Extensible Markup Language), and designed to be as compatible as possible with HTML, TEI (TEI, 2000), CES (CES, 2000), EAGLES (EAGLES, 2000), and LAL (Watanabe, 1999). It specifies modifier-modifiee relations, anaphor-referent relations, word senses, etc.

An example of a GDA-tagged sentence is as follows:

```
<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>
```

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

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. For instance, in the above sentence, <np rel="agt" sense="time0">Time</np> depends on the second

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

element `<v sense="fly1">flies</v>`. `rel="agt"` means that *Time* has the agent role with respect to the event denoted by *flies*.

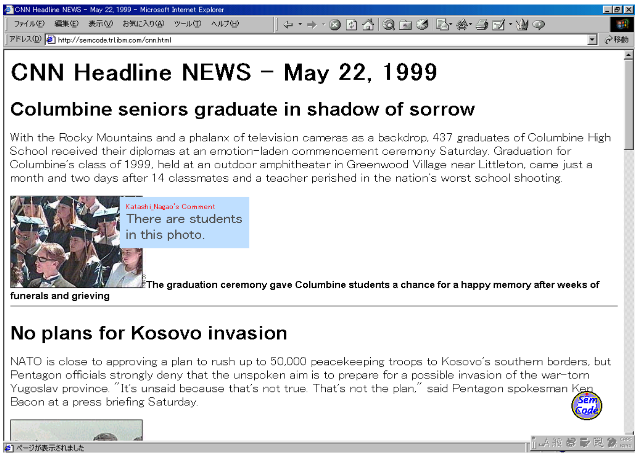
The `sense` attribute encodes a word sense.

Linguistic annotation is generated by automatic morphological analysis, interactive sentence parsing, and word sense disambiguation by selecting the most appropriate paraphrase. Some research issues on linguistic annotation are related to how the annotation cost can be reduced within some feasible levels. We have 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 accuracy of automatic annotation.

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

2.5 Commentary Annotation

Commentary annotation is mainly used to annotate non-textual elements like images and sounds with some additional information. Each comment can include not only tagged texts but also other images and links. Currently, this type of annotation appears in a subwindow that is overlayed on the original document window when a user locates a mouse pointer at the area of a comment-added element as shown in Figure 5.



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

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

Commentary annotaion on hyperlinks is also available. This contributes to quick introduction of target documents before clicking the links. If there are linguistic annotations on the target documents, the transcoders can generate summaries of these documents and relate them with hyperlinks in the source document.

There are some previous work on sharing comments on the Web. For example, ComMentor is a general meta-information architecture for annotating documents on the Web (Roscheisen et al., 1995). This architecture includes a basic client-server protocol, meta-information description language, a server system, and a remodeled NCSA Mosaic browser with interface augmentations to provide access to its extended functionality. ComMentor provides a general mechanism for shared annotations, which enables people to annotate arbitrary documents at any position in-place, share comments/pointers with other people (either publicly or privately), and create shared "landmark" reference points in the information space.

Such systems are often limited to particular documents or documents shared only among a few people. Our annotation and transcoding system can also handle multiple comments on any element of any document on the Web. Also, a community wide access control mechanism can be added to our transcoding proxy. If a user is not a member of a particular group, then the user cannot access the transcoding proxy that is for group use only. In the future, transcoding proxies and annotation servers will communicate with some secured protocol that prevents some other server or proxy from accessing the annotation data.

Our main focus is adaptation of WWW 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.

2.6 Multimedia Annotation

Our annotation technique can also be applied to multimedia data such as digital video. Digital video is becoming a necessary information source. Since the size of these collections is growing to huge numbers of hours, summarization is required to effectively browse video segments in a short time without losing the significant content. We have developed techniques for semi-automatic video annotation using a text describing the content of the video. Our techniques also use some video analysis methods such as automatic cut detection, characterization of frames in a cut, and scene recognition using similarity between several cuts.

There is another approach to video annotation. MPEG-7 is an effort within the Moving Picture Experts Group (MPEG) of ISO/IEC that is dealing with multimedia content description (MPEG, 2000).

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 the content publishers in guiding how multimedia content is transcoded in different situations and for different client devices. Also, MPEG-7 provides object-level description of multimedia content which allows a higher granularity of transcoding in which individual regions, segments, objects and events in image, audio and video data can be differentially transcoded 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 do not currently know when the MPEG-7 technology will be widely available.

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

Video annotation is performed through the following three steps.

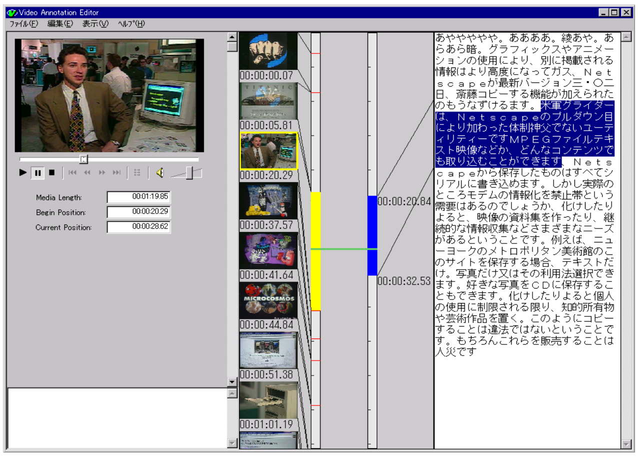
First, for each video clip, the annotation system creates the text corresponding to its content. We employed speech recognition for the automatic generation of a video transcript. The speech recognition module also records correspondences between the video frames and the words. The transcript is not required to describe the whole video content. The resolution of the description effects the final quality of the transcoding (e.g., summarization).

Second, some video analysis techniques are applied 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, frame sequences can be separated 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 some scene in order to reduce the time to identify target objects and to obtain a higher recognition success ratio. The user can name objects in a frame simply by selecting words in the corresponding text.

Third, the user relates video segments to text segments such as paragraphs, sentences, and phrases, based on scene structures and object-name correspondences. The system helps the user to 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, speech recognition, and correlation of scenes and words. An example screen of our video annotation editor is shown in Figure 6.



Media Length: 00:01:19.85
Begin Position: 00:00:20.29
Current Position: 00:00:28.62

あややややや。ああああ。綾あや。あらあら暗。グラフィックスやアニメーションの使用により、別に掲載される情報はより高度になってガス、Ｎｅｔｓｃａｐｅが最新バージョン三・〇二日、斎藤コピーする機能が加えられたのもうなずけるます。米軍グライダーは、Ｎｅｔｓｃａｐｅのプルダウン目により加わった体制神父でないユーティリティーですＭＰＥＧファイルテキスト映像などか、どんなコンテンツでも取り込むことができます、Ｎｅｔｓｃａｐｅから保存したものはすべてシリアルに書き込めます。しかし実際のところモデムの情報化を禁止帯という需要はあるのでしょうか、化けしたりよると、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。写真だけ又はその利用法選択できます。好きな写真をＣＤに保存することもできます。化けしたり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置く。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人災です

Figure 6: Video annotation editor

3 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 (HyperText Transfer Protocol) proxy server. Such an HTTP proxy is called a transcoding proxy.

Figure 7 shows the environment of semantic transcoding.

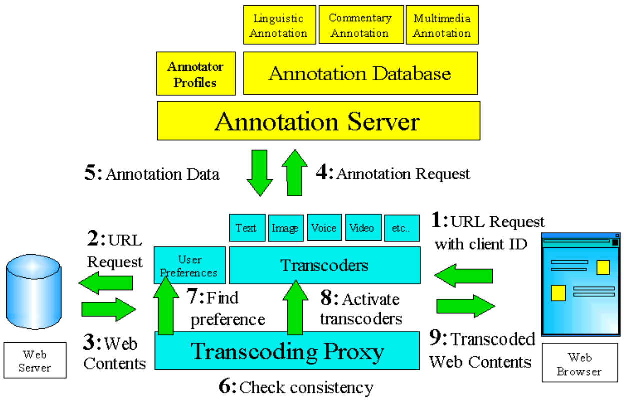


Figure 7: Transcoding environment

The information flow in transcoding is as follows:

1. The transcoding proxy receives a request URL with a client ID.
2. The proxy sends the request of the URL 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 annotation data of the URL 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 there is no preference data, 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.

3.1 Transcoding Proxy

We employed IBM's WBI (Web Intermediaries) as a development platform to implement the semantic transcoding system (WBI, 2000). WBI is a customizable and extendable HTTP proxy server. WBI provides APIs (Application Programming Interfaces) 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

3.2 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 (Nagao and Hasida, 1998). Since the method does not employ any heuristics dependent on the domain and style of documents, it is applicable to any linguistically-annotated document. The method can also trim sentences in the summary because importance scores are assigned to elements smaller than sentences.

A linguistically-annotated document naturally defines an intra-document network in which 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/anaphora links, document/subdivision/paragraph nodes, and rhetorical relation links.

The summarization algorithm works as follows:

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, the following elements are recursively marked as well, until no more elements are found:
   - the marker's head
   - the marker's antecedent
   - the marker's compulsory or *a priori* important daughters, the values of 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), etc.
   - the antecedent of a zero anaphor in the marker 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.

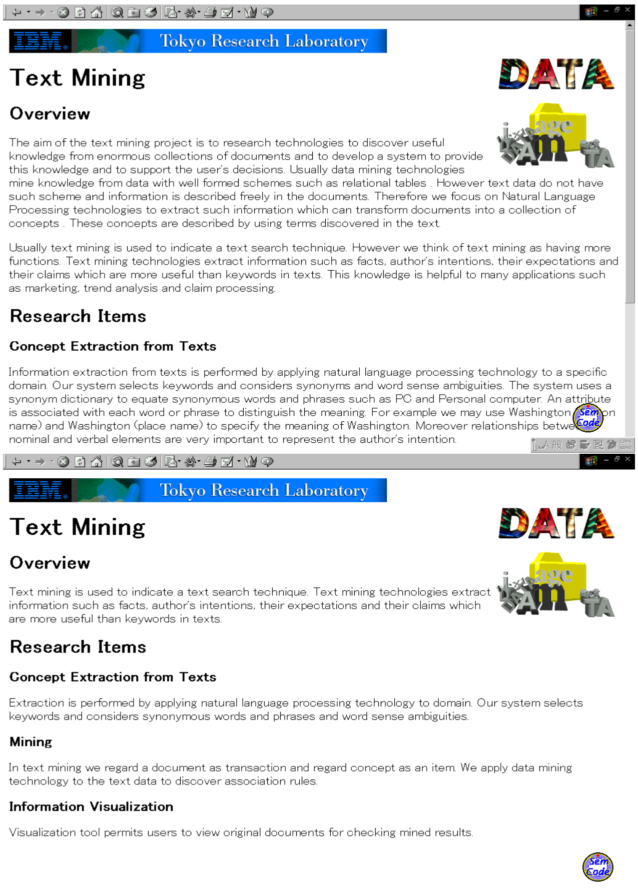
The size of the summary can be changed by simple user interaction. Thus the user can see the summary in a preferred size by using an ordinary Web browser without any additional software. The user can also input any words of interest. The corresponding words in the document are assigned numeric values that reflect degrees of interest. These values are used during spreading activation for calculating importance scores.

Figure 8 shows the summarization result on the normal Web browser. The top document is the original 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 can be easily clarified in annotation. An example of the result of English-to-Japanese translation is shown in Figure 9.

3.3 Image Transcoding

Image transcoding is to convert images into these of different size, color (full color or grayscale), and resolution (e.g., compression ratio) depending on user's



Tokyo Research Laboratory

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 specific domain. Our system selects keywords and considers synonyms and word sense ambiguities. The system uses 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 on name) and Washington (place name) to specify the meaning of Washington. Moreover relationships betwe nominal and verbal elements are very important to represent the author's intention.

Tokyo Research Laboratory

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

Extraction is performed by applying natural language processing technology to domain. Our system selects keywords and considers synonymous words and phrases and word sense ambiguities.

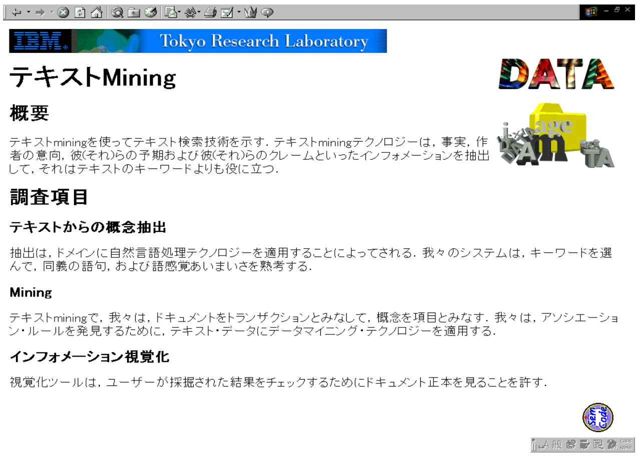
Mining

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

Information Visualization

Visualization tool permits users to view original documents for checking mined results.

Figure 8: Original and summarized documents



Tokyo Research Laboratory

テキストMining

概要

テキストminingを使ってテキスト検索技術を示す．テキストmining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

抽出は，ドメインに自然言語処理テクノロジーを適用することによってされる．我々のシステムは，キーワードを選んで，同義の語句，および語感覚あいまいさを熟考する．

Mining

テキストminingで，我々は，ドキュメントをトランザクションとみなして，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．

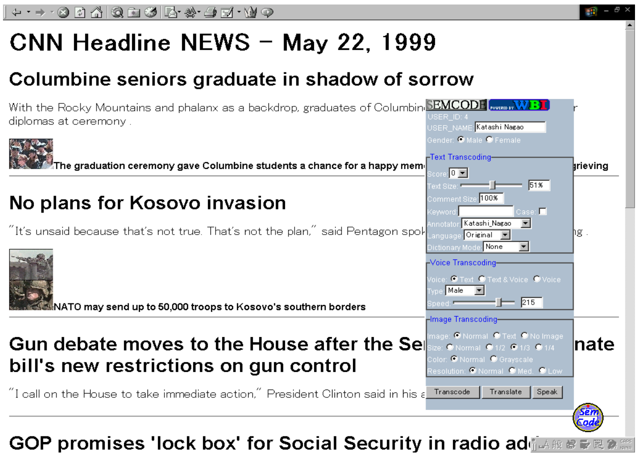
インフォメーション視覚化

視覚化ツールは，ユーザーが採掘された結果をチェックするためにドキュメント正本を見ることを許す．

Figure 9: Translated document

device and communication capability. Links to these converted images are made from the original images. Therefore, users will notice that the images they are looking at are not original if there are links to similar images.

Figure 10 shows the document that is summarized in half size of the original and whose images are reduced to one-third. In this figure, the preference setting subwindow is shown on the right hand. 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.



CNN Headline NEWS – May 22, 1999

Columbine seniors graduate in shadow of sorrow

With the Rocky Mountains and phalanx as a backdrop, graduates of Columbin r diplomas at ceremony .

The graduation ceremony gave Columbine students a chance for a happy mem grieving

No plans for Kosovo invasion

"It's unsaid because that's not true. That's not the plan," said Pentagon spok ng .

NATO may send up to 50,000 troops to Kosovo's southern borders

Gun debate moves to the House after the Se nate bill's new restrictions on gun control

"I call on the House to take immediate action," President Clinton said in his a

GOP promises 'lock box' for Social Security in radio ad

SEMCODE POWERED BY WBI

USER_ID: 4

USER_NAME: Katashi Nagao

Gender: ● Male ○ Female

Text Transcoding

Score: 0

Text Size: 51%

Comment Size: 100%

Keyword: Case: □

Annotator: Katashi_Nagao

Language: Original

Dictionary Mode: None

Voice Transcoding

Voice: ● Text ○ Text & Voice ○ Voice

Type: Male

Speed: 215

Image Transcoding

Image: ● Normal ○ Text ○ No Image

Size: ○ Normal ○ 1/2 ● 1/3 ○ 1/4

Color: ● Normal ○ Grayscale

Resolution: ● Normal ○ Med. ○ Low

Figure 10: Image transcoding (and preference setting window)

By combining image and text transcodings, the system can, for example, convert contents to just fit the client screen size.

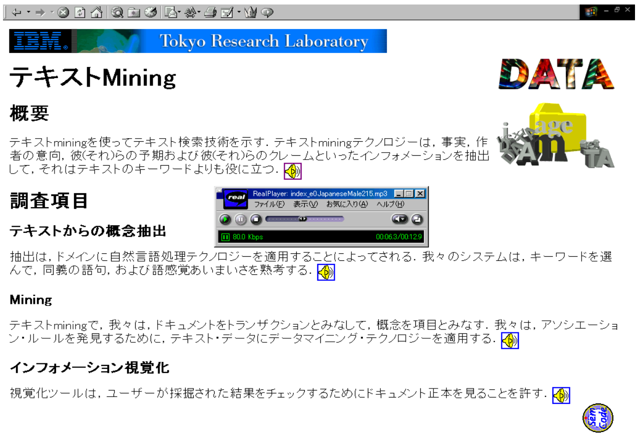
3.4 Voice Transcoding

Voice synthesis also works better if the content has linguistic annotation. For example, a speech synthesis markup language is being discussed in (SABLE, 2000). A typical example is processing proper nouns and technical terms. Word level annotations on proper nouns allow the transcoders to recognize not only their meanings but also their readings.

Voice transcoding generates spoken language version of documents. There are two types of voice transcoding. One is when the transcoder synthesizes sound data in audio formats such as MP3 (MPEG-1 Audio Layer 3). This case is useful for devices without voice synthesis capability such as cellular phones and PDAs. The other is when the transcoder converts documents into more appropriate style for voice synthesis. This case requires that a voice synthesis program is installed on the client side. Of cource, the synthesizer uses the output of the voice synthesizer. Therefore, the mechanism of document conversion is a common part of both types of voice transcoding.

Documents annotated for voice include some text in commentary annotation for non-textual elements and some word information in linguistic annotation for the reading of proper nouns and unknown words in the dictionary. The document also contains phrase and sentence boundary information so that pauses appear in appropriate positions.

Figure 11 shows an example of the voice-transcoded document in which icons that represent the speaker are inserted. When the user clicks the speaker icon, the MP3 player software is invoked and starts playing the synthesized voice data.


Tokyo Research Laboratory

テキストMining

概要

テキストminingを使ってテキスト検索技術を示す．テキストmining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

抽出は，ドメインに自然言語処理テクノロジーを適用することによってされる．我々のシステムは，キーワードを選んで，同義の語句，および語感覚あいまいさを熟考する．

Mining

テキストminingで，我々は，ドキュメントをトランザクションとみなして，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

視覚化ツールは，ユーザーが採掘された結果をチェックするためにドキュメント正本を見ることを許す．

RealPlayer: index_e0JapaneseMale215.mp3

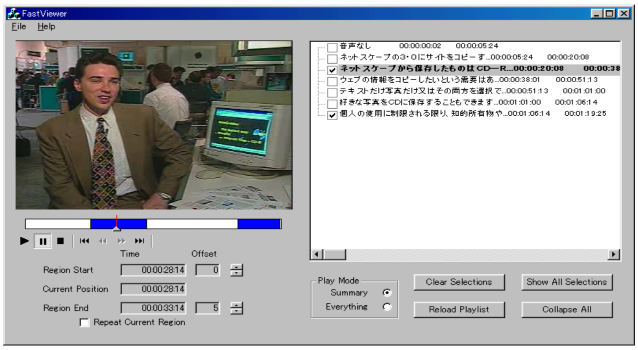
Figure 11: Voice transcoding

3.5 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, etc. Our video transcoding has several variations, including video summarization, video to document transformation, video translation, etc.

Video summarization is performed 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. Summarized video is played by a player we developed. An example screen of our video player is shown in Figure 12.

There are some previous work on video summarization such as Infomedia (Smith and Kanade, 1995) and CueVideo (Amir et al., 1999). They 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. They can transcode video data without annotations. However, currently, an accuracy of their summarization is not practical because of the failure of automatic video analysis. Our approach to video summarization has sufficient quality for use if the data has enough semantic annotation. As mentioned earlier, we have developed a tool to help annotators to 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.


FastViewer

□ 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 00:00:20:08
☑ **ネットスケープから保存したものはCD—R...00:00:20:08 00:00:38**
□ ウェブの情報をコピーしたいという需要はあ...00:00:38:01 00:00:51:13
□ テキストだけ写真だけ又はその両方を選択で...00:00:51:13 00:01:01:00
□ 好きな写真をCDに保存することもできます...00:01:01:00 00:01:06:14
☑ 個人の使用に制限される限り、知的所有物や...00:01:06:14 00:01:19:25

	Time	Offset
Region Start	00:00:28:14	0
Current Position	00:00:28:14	
Region End	00:00:33:14	5

□ Repeat Current Region

Play Mode
Summary ●
Everything ○

Figure 12: Video player with summarization function

Video to document transformation is another type of video transcoding. If the client device does not have video playing capability, the user cannot access video contents. In this case, the video transcoder creates a document including important images of scenes and texts related to each scene. Also, the resulting document can be summarized by the text transcoder.

Video translation is a combination of text and voice transcodings. First, a video transcript with linguistic annotation is translated by the text transcoder. Then, the result of translation is converted into voice-suitable text by the voice transcoder. Synchronization of video playing and voice synthesis makes another language version of the original video clip. This part has not yet been implemented, but this function will be integrated into our video player.

The above described transcodings are automatically combined according to user demand, so the transcoding proxy has a planning machanism to determine the order of activation of each transcoder necessary for the requested content and user preferences (including client device constraints).

4 Future Plans

We are planning to apply our technology to knowledge discovery from huge online resources. Annotations will be very useful to extract some essential points in documents. For example, an annotator adds comments to several documents, and he or she seems to be a specialist of some particular field. Then, the machine automatically collects documents annotated by this annotator and generates a single document including summaries of the annotated documents.

Also, content-based retrieval of Web documents including multimedia data is being pursued. Such retrieval enables users to ask questions in natural language (either spoken or written).

While our current prototype system is running locally, we are also planning to evaluate our system with some open experiments.

5 Concluding Remarks

We have discussed a full architecture for creating and utilizing external annotations. Using the annotations, we realized semantic transcoding that automatically customizes Web contents depending on user preferences.

This technology also contributes to commentary information sharing and device dependent transformation for any device. One of our future goals is to make contents of the WWW intelligent enough to answer our questions asked using natural language. We imagine that in the near future we will not use search engines but will instead use knowledge discovery engines that give us a personalized summary of multiple documents instead of hyperlinks. The work in this paper is one step toward a better solution of dealing with the coming information deluge.

Acknowledgments

The authors would like to acknowledge the contributors to the project described in this paper. Koiti Hasida gave the authors some helpful advices to apply the GDA tag set to our linguistic and multimedia annotations. Hideo Watanabe developed the prototype of English-to-Japanese language translator. Shinichi Torihara developed the prototype of voice synthesizer.

References

A. Amir, S. Srinivasan, D. Ponceleon, and D. Petkovic. CueVideo: Automated indexing of video for searching and browsing. In *Proceedings of SIGIR'99*. 1999.

Corpus Encoding Standard (CES). Corpus Encoding Standard. http://www.cs.vassar.edu/CES/.

Expert Advisory Group on Language Engineering Standards (EAGLES). EAGLES online. http://www.ilc.pi.cnr.it/EAGLES/home.html.

Koiti Hasida. Global Document Annotation. http://www.etl.go.jp/etl/nl/gda/.

IBM Almaden Research Center. Web Intermediaries (WBI). http://www.almaden.ibm.com/cs/wbi/.

Hiroshi Maruyama, Kent Tamura, and Naohiko Uramoto. XML and Java: Developing Web applications. Addison-Wesley, 1999.

Moving Picture Experts Group (MPEG). MPEG-7 Context and Objectives. http://drogo.cselt.stet.it/mpeg/standards/mpeg-7/mpeg-7.htm.

Katashi Nagao and Koiti Hasida. Automatic text summarization based on the Global Document Annotation. In *Proceedings of COLING-ACL'98*. 1998.

Martin Roscheisen, Christian Mogensen, and Terry Winograd. Shared Web annotations as a platform for third-party value-added information providers: Architecture, protocols, and usage examples. *Technical Report CSDTR/DLTR*. Computer Science Department, Stanford University, 1995.

The SABLE Consortium. A Speech Synthesis Markup Language. http://www.cstr.ed.ac.uk/projects/ssml.html.

Michael A. Smith and Takeo Kanade. Video skimming for quick browsing based on audio and image characterization. *Technical Report CMU-CS-95-186*. School of Computer Science, Carnegie Mellon University, 1995.

The Text Encoding Initiative (TEI). Text Encoding Initiative. http://www.uic.edu:80/orgs/tei/.

Hideo Watanabe. Linguistic Annotation Language: The markup langauge for assisting NLP programs. *TRL Research Report RT0334*. IBM Tokyo Research Laboratory, 1999.

World Wide Web Consortium. Extensible Markup Language (XML). http://www.w3.org/XML/.

World Wide Web Consortium. XML Path Language (XPath) Version 1.0. http://www.w3.org/TR/xpath.html.

From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools

M. Erdmann, A. Maedche, H.-P. Schnurr, S. Staab
Institute AIFB, Karlsruhe University, 76128 Karlsruhe, Germany
{erdmann, maedche, schnurr, staab}@aifb.uni-karlsruhe.de
http://www.aifb.uni-karlsruhe.de/WBS

Abstract

Semantic Annotation is a basic technology for intelligent content and is beneficial in a wide range of content-oriented intelligent applications. In this paper we present our work in ontology-based semantic annotation, which is embedded in a scenario of a knowledge portal application. Starting with seemingly good and bad manual semantic annotation, we describe our experiences made within the KA^2-initiative. The experiences gave us the starting point for developing an ergonomic and knowledge base-supported annotation tool. Furthermore, the annotation tool described are currently extended with mechanisms for semi-automatic information-extraction based annotation. Supporting the evolving nature of semantic content we additionally describe our idea of evolving ontologies supporting semantic annotation.

1 Introduction

The KA^2-initiative (Knowledge Annotation initiative of the Knowledge Acquisition community) was launched at EKAW in 1997 in order to provide semantic access to information stored in web pages in the WWW. It built on manual semantic annotation for integration and retrieval of facts from semantically annotated web pages, which belonged to members of the knowledge acquisition community (Decker et al., 1999; Benjamins et al., 1999). The initiative recently developed into a more comprehensive concept viz. the KA^2 community portal, which allows for providing, browsing and retrieving information through various means of ontology-based support (Staab et al., 2000). All along the way, the usage of semantic annotation as the underpinning for semantics-based fact retrieval, integration, and presentation has remained one of the major cornerstones of the system.

The content of the paper is organized as follows. In Section 2 we start with a brief introduction to our notion of a community web portal to set up the context of our use of semantic annotations. Then, we present the practical problems we have encountered with manual annotations and the lessons learned from these experiences (cf. Section 3). In Section 4 the development of annotation tools is sketched that facilitate manual semantic annotation by following ergonomic considerations about the process that someone who is annotating information goes through and inferencing support that provides a comprehensive view on what has been annotated, so far. The development of an information extraction-based system for semi-automatic annotation that proposes annotations to the human who is performing annotations is presented in Section 5. We conceive semantic annotation as a cyclic process between the actual task of annotating documents and the development and adaptation of a *domain ontology*. Incoming information that is to be annotated does not only require some more annotating, but also continuous adaptation to new semantic terminology and relationships. This cyclic process of evolving ontologies is shown in Section 6. Our objective here is to give the reader a comprehensive picture of what semantic annotation has meant in our application and where it is heading now.

2 Scenario: Semantic Community Web Portal

Community web portals serve as high quality information repositories for the information needs of particular communities on the web. A prerequisite for fulfilling this role is the accessibility of information. In *community* portals this information is typically provided by the users of the portal, i.e. the portal is driven *by* the community *for* the community. We have been maintaining a web portal for the Knowledge Acquisition community[1] and, thus, have gained some experience with the difficulties of providing information for that portal by semantic annotations.

We here give only a very brief sketch of the KA community web portal. A broader introduction to the methods and tools developed in this context can be found in (Staab et al., 2000). The portal's main component is Ontobroker (Decker et al., 1999), that uses ontologies to provide an integrated view on distributed, heterogenous information sources. The ontology is the means for capturing domain knowledge in a generic way that provides a commonly agreed understanding of a domain, which may be reused and shared within communities or applications. The ontology can be used to semantically annotate web pages that are accessed by Ontobroker

The Ontobroker system consists of (i) a crawling component, (ii) a knowledge base, (iii) an inference engine, and (iv) a query interface. The crawler collects informa-

[1] `http://ka2portal.aifb.uni-karlsruhe.de`

tion contained in registered web pages and stores it in the knowledge base. The HTML pages are manually annotated with special semantic tags, a proprietary extension to HTML that is compatible with common web browsers. This annotation language is presented in the next section. Thus, the web crawler establishes the core of the knowledge base, that is enhanced by applying axioms from the ontology to these ground facts. The ontology is represented in Frame Logic (Kifer et al., 1995), an object-oriented and logics-based language. Thus, axioms can be formulated using a subset of first order logic statements including object oriented modelling primitives. Finally, the information stored in the knowledge base or derived by the inference engine can be accessed using Frame Logic queries.

3 Manual Semantic Annotations

3.1 HTML-A

The main source of information for the KA portal stems from distributed web pages maintained by members of the KA community. These web pages have been manually annotated to explicitely represent the semantics of their contents (cf. Figure 1). Since a huge amount of relevant information for most communities is represented in HTML, we chose to enhance HTML with few semantically relevant extensions. The resulting annotation language HTML-A (Decker et al., 1999) adds to HTML primitives for tagging instances of concepts, for relating these instances, and for setting their properties, i.e. the ontology serves as a schema for semantic statements in these pages. For all these primitives the HTML anchor tag `<A>` has been extended with a special attribute `onto`. This decision implies that the original information sources hardly have to be changed to provide semantically meaningful information. The semantic tags are embedded in the ordinary HTML text in such a way that standard browsers can still process the HTML pages and, at the same time, Ontobroker's crawler can extract the semantic annotations from them. This kind of semantic annotation resembles Knuth's literate programming (Knuth, 1984), where few semantically relevant and formal statements are embedded in unstructured prose text. In Ontobroker, objects (instances of concepts) are uniquely identified by a URI, i.e. resources in the web are interpreted as surrogates for real objects like persons, organizations, and publications. To associate (in HTML) such an object with a concept from the ontology one of the following statements can be made in the HTML source.

```
<A onto="'http://www.aifb.uni-
          karlsruhe.de/studer':Researcher"></A>
<A onto="'www9':InProceedings"></A>
<A onto="page:Institute"></A>
```

In the schema `<A onto="`O`:`C`"></A>` of these expressions O represents the instance and C represents the concept. O can either be a global URI, a local part of a URI (that is expanded by the crawler to a global one), or one of the special keywords `page`, `body`, `href`, or `tag`. These special keywords represent resources relative to the current tag and the current web page, e.g. the keyword `page` represents the URI of the webpage of this statement. The following statements both define formally the value of the `name` attribute of the object represented by the current `page`:

```
<A onto="page[name='Rudi Studer']"></A>
<A onto="page[name=body]">Rudi Studer</A>
```

The keyword `body` refers to the content of the anchor tag. Thus, the actual information is rendered by a web browser and at the same time interpreted formally by the crawler. Including semantics in this way into HTML pages reduces redundancy and enhances maintainability, since changes in the prose part of the page are immediately reflected in the formal part, as well.

To establish relationships between two objects similar statements can be made, since binary relations can be modelled as attributes:

```
<A onto="page[affiliation='http://www.aifb.uni-
                           karlsruhe.de']"></A>
<A onto="page[affiliation=href]"
         href="http://www.aifb.uni-karlsruhe.de">
        Institut AIFB</A>
<A onto="page[authorOf=href]"
         href="publications.html#www9">
        Semantic Community Web Portals</A>
```

The `href` keyword defines the target of the hypertext link as an object representing the value of the attribute. If this link is relative it is expanded to its global URI before putting the facts into the knowledge base.

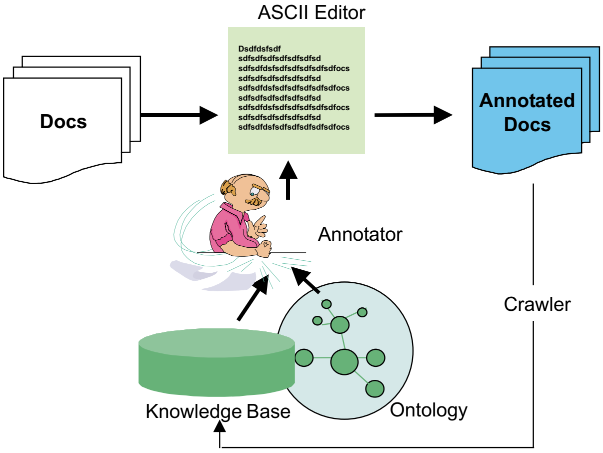


Figure 1: Manually annotating HTML pages with semantic information.

3.2 Experiences

Our experiences with the KA2 initiative were quite disappointing, concerning the information providing process. There were about 30 people willing to provide information from their web pages to Ontobroker. About 15 accepted and were (more or less) able to annotate their pages. The other 15 needed rather extensive support from the Ontobroker team. One of our students prepared annotated versions of their homepages. Since the annotation task was not supported by any tool, severe problems appeared. First of all, a lot of annotations were simply syntactically incorrect, i.e. have been rejected by the parser, for reasons like missing brackets or quotation marks. This problem has been remedied by providing a syntax checker that is available online and tests annotated pages for syntactic correctness.

A second major problem concerned terminology. Since the ontology was fixed from the beginning, the annotations had to strictly conform to the concept and attribute names defined in the ontology. Typing errors, e.g. `online-Version` instead of `onlineVersion`, were the most prominent in this category.

The last group of problems deals with the semantics of the annotations:

- The class of some objects had been defined in a too general manner, e.g. most publications in the KA2 knowledge base have been categorized simply as `Publication` instead of `JournalArticle`, `TechnicalReport` or another more specific concept. The intention of some ontological terms have not been completely understood by some providers. This resulted in things like the classification of a web page containing a list of publications of some researcher to be defined as the value of his `publication` attribute. This set valued attribute was intended to contain a set of `publication` objects and not a single container object. Application of axioms in the ontology yielded the fact that this publication list was categorized as a `Publication` which was not intended. On the other hand, querying the knowledge base for all publications of this researcher resulted in an acceptable answer, namely a link to this list page. Additionally, each object should be identified by a single URI. But we experienced major problems with the use of object identifiers to refer to certain objects in an unambiguous way.
- Often, instead of introducing a URI or referring to an existing object identifier to denote an object, information providers simply used text from the web page, e.g. a co-author of a publication was often identified by a string like "John Doe" instead of the URI for his home page.
- Similarly, even if object identifiers, i.e. URIs, were used to refer to remote objects like co-authors, these URIs often did not match. An implication of these mismatches is the creation of several objects that should have been unified into one, e.g. our colleague Dieter Fensel at some time was represented in the knowledge base by three object identifiers, each denoting an object with some information linked to it. These information could be integrated only after the sources of the mismatch had been identified.
- Finally, the overall quantity of semantic annotations could have been larger, i.e. although some information was textually present on the annotated web pages, this information has not been annotated and, thus, was invisible for Ontobroker and for its users. This problem especially occured on pages annotated by our student, due to her lack of deep domain knowledge.

3.3 Lessons learned

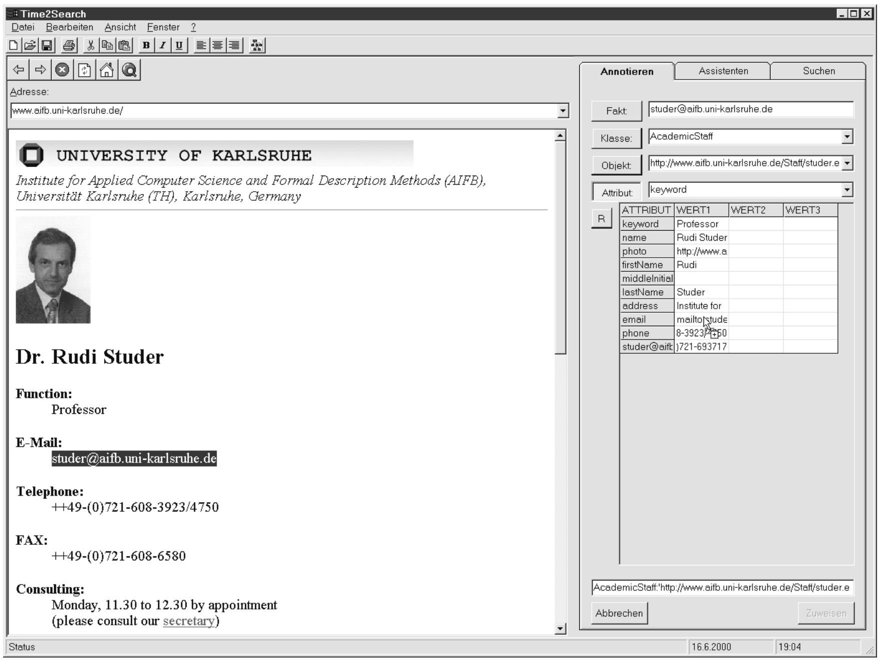
After reviewing the different types of problems, we came up with a set of lessons learned that may be summarized by:

- Keep the ontology simple and explain its meaning!
- Support annotators with interactive, graphical tools!
  - to help avoid syntax errors and typos of ontological entities,
  - to help to correctly choose the most specific concepts for instances, and
  - to provide a list of all known objects of a certain concept to reduce false co-references (with a kind of repository of objects)
- Allow importing information from other sources to avoid annotations where possible, e.g. import BiBTeX-files for the publications of researchers.

4 Ergonomic and knowledge base-supported Annotation

Targeting to an ergonomic and intuitive support of the annotation task within documents, we developed the annotation tool. It allows the quick annotation of facts within any document by tagging parts of the text and semantically defining its meaning via interacting with the dialog shown in the screenshot of Figure 2. To illustrate the annotation process using the annotation tool, we sketch in the following a small annotation scenario using the annotation tool.

Given an ontology, the annotation process usually starts with tagging one or more phrases in the document, an HTML file in our example. This selection is indicated in the fact (FAKT) field in the right column of Figure 2. The user selects the appropriate concept in the ontology, depicted in the KLASSE field in the right column of Figure 2 as an explorer tree view. In our example, `studer@aifb.uni-karlsruhe.de` is chosen and the concept AcademicStaff is selected in the ontology. Therefore, the annotation tool supports the intuitive and correct choice of the most specific concept for the selected instance. Now, as described in further detail in (Schnurr and Staab, 2000) the concept choice of the user triggers the F-Logic inference engine to search for all known objects of this certain concept in the knowledge base. The third row (OBJEKT) in the right column in Figure 2 shows these objects, in our example a list of objects of the concept AcademicStaff. Thus, the user may insert references to the known objects or add a new object to the knowledge base. In our example, the user selects `http://www.aifb.uni-karlsruhe.de/Staff/studer.en.html`, the primary key of the researcher with the last name `Studer`,



Adresse: www.aifb.uni-karlsruhe.de/

UNIVERSITY OF KARLSRUHE

Institute for Applied Computer Science and Formal Description Methods (AIFB), Universität Karlsruhe (TH), Karlsruhe, Germany

Dr. Rudi Studer

Function:
Professor

E-Mail:
studer@aifb.uni-karlsruhe.de

Telephone:
++49-(0)721-608-3923/4750

FAX:
++49-(0)721-608-6580

Consulting:
Monday, 11.30 to 12.30 by appointment
(please consult our secretary)

Fakt: studer@aifb.uni-karlsruhe.de

Klasse: AcademicStaff

Objekt: http://www.aifb.uni-karlsruhe.de/Staff/studer.e

Attribut: keyword

ATTRIBUT	WERT1	WERT2	WERT3
keyword	Professor		
name	Rudi Studer		
photo	http://www.a		
firstName	Rudi		
middleInitial			
lastName	Studer		
address	Institute for		
email	mailto:stude		
phone	8-3923/4750		
studer@aifk	)721-693717		

AcademicStaff:'http://www.aifb.uni-karlsruhe.de/Staff/studer.e

16.6.2000 19:04

Figure 2: Annotation Dialog.

to add his EMAIL address to the knowledge base. If there would be no corresponding known object in the knowledge base, the user would have to select AcademicStaff_Neu to add a new instance. Thereby, the system automatically creates a primary key for that new object. In the middle of the right column of Figure 2, the attributes of the highlighted concept are shown. The selected part of the document, namely studer@aifb.uni-karlsruhe.de in our example, may now be moved via drag-and-drop to the appropriate attribute, in our example the attribute EMAIL. The user thereby annotates the selected part of the document. Clicking the ”R”-button in the middle of the right column in Figure 2 shows a list of relations linked to the chosen concept. Selecting one of the indicated relations gives a list of possible instances, where the relation may point to. The user picks up the appropriate instance and thus, links both concepts with the selected relation. The dialog shown in Figure 2 offers the whole range of annotation support to the user. With the features of our annotation tool, we support annotators with an interactive, graphical means helping to avoid syntax errors. We support them in choosing the most appropriate concepts for instances and provide an object repository to identify existing instances. As indicated in Figure 3, the annotation tool integrates the ontology and the knowledge base into the editing environment to allow for ergonomic and knowledge base-supported annotating.

5 Semi-Automatic Annotation

Based on our experiences and the existing annotation tool for supporting ontology-based semantic annotation

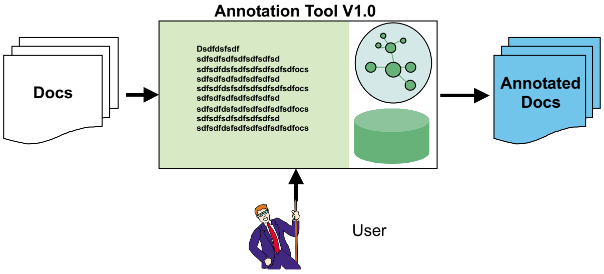


Figure 3: Ergonomic and inference-supported Annotation.

of texts, we now approach semi-automatic annotation of natural language texts. We conceive an information extraction-based appraoch for semi-automatic annotation, which has been implemented on top of SMES (Saarbrücken Message Extraction System), a shallow text processor for German (cf. (Neumann et al., 1997)). This is a generic component that adheres to several principles that are crucial for our objectives. *(i)*, it is fast and robust, *(ii)*, it realizes a mapping from terms to ontological concepts, *(iii)* it yields dependency relations between terms, and, *(iv)*, it is easily adaptable to new domains.[2]

We here give a short survey on SMES in order to provide the reader with a comprehensive picture of what underlies our system. The architecture of SMES comprises a *tokenizer* based on regular expressions, a *lexical anal-*

[2] The interlinkage between the information extraction system SMES and domain ontologies is described in further detail in (Staab et al., 1999).

ysis component including a *word and a domain lexicon*, and a *chunk parser*. The tokenizer scans the text in order to identify boundaries of words and complex expressions like "$20.00" or "Mecklenburg-Vorpommern"[3], and to expand abbreviations. The lexicon contains more than 120,000 stem entries and more than 12,000 subcategorization frames describing information used for lexical analysis and chunk parsing. Furthermore, the domain-specific part of the lexicon associates word stems with concepts that are available in the concept taxonomy. *Lexical Analysis* uses the lexicon to perform, *(1)*, morphological analysis, *i.e.*, the identification of the canonical common stem of a set of related word forms and the analysis of compounds, *(2)*, recognition of name entities, *(3)*, retrieval of domain-specific information, and, *(4)*, part-of-speech tagging. While the steps (1),(2) and (4) can be a viewed as standard for information extraction approaches (cf. (Appelt et al., 1993; Neumann et al., 1997)), the step (3) is of specific interest for our annotation task. This step associates single words or complex expressions with a concept from the ontology if a corresponding entry in the domain-specific part of the lexicon exists. E.g., the expression "Hotel Schwarzer Adler" is associated with the concept Hotel.

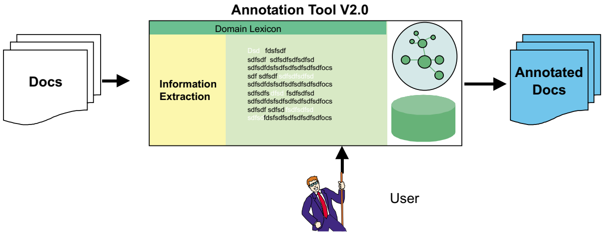


Figure 4: Semi-automatic Annotation.

SMES includes a *chunk parser* based on weighted finite state transducers to efficiently process phrasal and sentential patterns. The parser works on the phrasal level, before it analyzes the overall sentence. Grammatical functions (such as subject, direct-object) are determined for each dependency-based sentential structure on the basis of subcategorizations frames in the lexicon. Our primary output derived from SMES consists of *dependency relations* (Hudson, 1990) found through lexical analysis (compound processing) and through parsing at the phrase and sentential level. Thereby, the grammatical dependency relation need not even hold directly between two conceptually meaningful entities. For instance, in the sentence "The `Hotel Schwarzer Adler` in `Rostock` celebrates Christmas.", "Hotel Schwarzer Adler" and "Rostock", the concepts of which appear in the ontology as Hotel and City, respectively, are not directly connected by a dependency relation. However, the preposition "in" acts as a mediator that incurs the conceptual pairing of Hotel with City.

Figure 4 depicts the architecture of the semi-automatic annotation tool. Incoming documents are processed using the information extraction system SMES. SMES associates single words or complex expressions with a concept from the ontology, connected through the domain lexicon linkage. Recognized concepts and dependency relations between concepts are highlighted as suggested annotations. This mechanism has the advantage that all relevant information in the document with regard to the ontology is recognized and proposed to the annotator. The actual process of annotation is delegated to the annotation tool described in section 4.

6 Evolving Ontologies

In the previous sections 3, 4 and 5 we have abstracted from the interlinkage between evolving ontologies and the different annotation mechanisms. However, in any realistic application scenario, incoming information that is to be annotated does not only require some more annotating, but also continuous adaptation to new semantic terminology and relationships. Terms evolve in their meanings, or take on new meanings as new technologies are developed, and as existing ones evolve.

The abstraction from the interlinkage between annotation and evolving ontologies resulted in problems, *(i)* if the meaning of ontological elements changed, *(ii)* if the elements in the ontology became unnecessary and have been eliminated, or *(iii)* if new elements have been added to the ontology. Our experiences have shown that annotation and ontology development and maintenance must be considered as a cyclic process. Thus, in a realistic annotation scenario a feedback loop and tight integration is required, so that new conceptual structures can be added to the ontology for supporting the actual task of annotating documents towards evolving ontologies.

Manual Ontology Engineering. Starting with manual semantic annotation as described in Section 3 the ontology was represented as an ASCII file in FLogic. There was only few documentation, no browsing was possible, and it was fixed from the beginning. The process of manual semantic annotation didn't incorporate the ontology, so that typing errors were not unusual. One of the more fundamental problems were incorrect coreferences, because no interlinkage between new annotated facts and existing facts was supported.

As described in Section 4 our experiences showed us the necessity for ergonomic and knowledge base-supported annotation. We developed a tool which includes the domain ontology directly in its interface, defines automatically identifiers and references to existing facts contained in the knowledge base. We also developed an ontology engineering environment OntoEdit[4] supporting the ontology engineer in modeling conceptual structures.

Semi-Automatic Ontology Engineering. Currently we are working on the tight integration between seman-

[3] Mecklenburg-Vorpommern is a region in the north east of Germany.

[4] A comprehensive description of the ontology engineering environment OntoEdit and the underlying methodology is given in (Staab and Maedche, 2000).

tic annotation and ontology engineering. Lexical resources are directly mapped onto concepts and relations contained in the ontology. The coding nature of ontologies makes it necessary to account for changes. Hence, we have been developing methods that propose new conceptual structures to the maintainer of the ontology (cf. (Maedche and Staab, 2000a)). In parallel, linguistic resources are gathered, which connect the conceptual structures with the information extraction system. The information extraction system supports the engineering of evolving ontologies as well as the process of extracting annotation-relevant information. The underlying idea is that acquired domain specific knowledge and linguistic resources are connected to natural language using a tight interplay between ontology and domain lexicon.

In (Maedche and Staab, 2000b) we describe our work in semi-automatic engineering and learning of domain ontologies from text. A comprehensive architecture lays the foundation for acquiring domain ontologies and linguistic resources. The main components of the architecture are *(i)* the Text & Processing Management, *(ii)* the Information Extraction Server (SMES), *(iii)* a Lexical Database and Domain Lexicon, *(iv)* a Learning Module, and *(v)* the Ontology Engineering Environment OntoEdit. The architecture has been fully implemented in the “Ontology Learning”-Environment Text-To-Onto and lays the foundation for supporting the development of evolving ontologies from text.

7 Related Work

An approach similar to our first tries of annotating HTML using ontologies has been developed at the University of Maryland. The SHOE system (Luke et al., 1997) defines additional tags that can be embedded in the body of HTML pages. In SHOE there is no direct relationship between the new tags and the original text of the page, i.e. SHOE tags are not annotations in a strict sense. In (Heflin et al., 1999), the authors report of similar observations of the “annotation” process as we present here.[5]

When talking about semantic annotations, terms like XML (Bray et al., 1998) and RDF (Lassila and Swick, 1999) must not be absent. Especially XML (Extensible Markup Language) earned a lot of attention in the last two years since its standardisation. XML allows the definition of individual tags that can be interpreted according to the user’s will. E.g. XHTML represents an HTML-like vocabulary to describe the layout of web pages for browsers, SMIL defines tags that describe complete multimedia documents, or with XMLNews-tags the text of news can be annotated with rich semantic meaning such as the location and date of an event. Pure XML vocabularies like these are not sufficient as means for representing deep semantics, but they can be complemented by ontologies to achieve a flexible and well understood way to represent and transfer content (via XML) and at the same time to embed the represented facts in a formal and machine interpretable model of discourse (via the ontology). In (Erdmann and Studer, 1999) we show how to establish such a close coupling automatically.

We expect the relationship of semantic annotations or semantic metadata with ontologies to be central for the success of semantic information processing in the future. The Resource Description Framework (RDF), an (XML-based) representation format for meta data defined by the W3C could take a central part in this development, since an ontology representation mechanism has been defined on top of the basic RDF primitives. A core language introducing notions of classes and relationships has been proposed to the W3C as RDFS (Brickley and Guha, 1999). Even richer languages for more elaborate modeling primitives like symmetric relationships, part-of relations, or Description-Logic-like subsumption hierarchies were proposed in (Erdmann et al., 2000) or (Horrocks et.al., 2000). Thus, RDF could become *the means* to represent metadata *and* ontologies in an open, widely “spoken” representation and interchange format.

Concerning our mechanisms for semi-automatic semantic annotation described in Section 5 there has been done only little research. Pustejovsky et al. (Pustejovsky et al., 1997) describe their approach for semantic indexing and typed hyperlinking. As in our approach finite state technologies support lexical acquisition as well as semantic tagging. The goal of the overall process is the generation of so called *lexical webs* that can be utilized to enable automatic and semi-automatic construction of web-based texts.

In (Bod et al., 1997) approaches for learning syntactic strctures from syntactically tagged corpus has been transferred to the semantic level, too. In order to tag a text corpus with type-logical formulae, they created tool environment called SEMTAGS for semi-automatically enriching trees with semantic annotations. SEMTAGS incrementally creates a first order markov model based on existing annotations and proposes a semantic annotation of new syntactic trees. The authors report promising results: After the first 100 sentences of the corpus had been annotated, SEMTAGS already produced the correct annotations for 80% of the nodes for the immediately subsequent sentences.

8 Discussion

Based on the KA^2 community portal scenario we have shown in Section 3 how information has been provided in the beginning. Our lessons learned from this experience gave us a starting point for developing more advanced and more user friendly methods for semantically annotating documents. The methods are combined with an information extraction system that semi-automatically proposes new annotations to the user. Our experiences have shown that semantic annotation and ontology engineering must be considered a cyclic process.

In the future much work remains to be done. First,we will have to build an integrated system of annotation and ontology construction. This system will combine

[5] For a further comparison of several ways to represent knowledge in the WWW (often by means like semantic annotations) refer to (van Harmelen and Fensel, 1999).

knowledge base-supported, ergonomic annotation, with an environment and methods for ontology engineering and learning from text supporting evolving ontologies. Second, we have to evaluate our annotation mechanisms. Evaluation in our annotation architecture can be splitted into several sub-evaluation phases: ergonomic evaluation, evaluation of the ontology, evaluation of the semi-automatic suggestions, evaluation of the user's annotations. Third, we will support the RDF standard for representing metadata on the web, representing both ontologies and generated annotated facts in RDF(S). This standard will make annotated facts reusable and machine-processable on the web (Decker et al., 2000).

Acknowledgements. We thank Stefan Decker for initiating the first version of an annotation tool. We thank our students Mika Maier-Collin and Jochen Klotzbuecher for the Annotation Tool; Dirk Wenke for the Ontology Engineering Environment OntoEdit; Raphael Volz for the Ontology Learning Environment Text-To-Onto; DFKI language technology group, in particular Guenter Neumann, for their dedicated support in using SMES. This work was partially supported by grant GETESS and by Ontoprise GmbH.

References

D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. 1993. FASTUS: A finite state processor for information extraction from real world text. In *Proceedings of IJCAI-93*, Chambery, France, August.

R. Benjamins, D. Fensel, and S. Decker. 1999. KA2: Building Ontologies for the Internet: A Midterm Report. *International Journal of Human Computer Studies*, 51(3):687.

R. Bod, R. Bonnema, and R. Scha. 1997. Data-oriented semantic interpretation. In *In Proceedings of the Second International Workshop on Computational Semantics (IWCS), Tilburg, 1997*.

T. Bray, J. Paoli, and C.M. Sperberg-McQueen. 1998. Extensible markup language (XML) 1.0. Technical report, W3C. http://www.w3.org/TR/1998/REC-xml-19980210.

D. Brickley and R.V. Guha. 1999. Resource description framework (RDF) schema specification. Technical report, W3C. W3C Proposed Recommendation. http://www.w3.org/TR/PR-rdf-schema/.

S. Decker, M. Erdmann, D. Fensel, and R. Studer. 1999. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In R. Meersman et al., editors, *Database Semantics: Semantic Issues in Multimedia Systems*, pages 351–369. Kluwer Academic Publisher.

S. Decker, J. Jannink, P. Mitra, S. Staab, R. Studer, and G. Wiederhold. 2000. An information food chain for advanced applications on the www. In *Proceedings of the Fourth European Conference on Research and Advanced Technology for Digital Libraries*.

M. Erdmann and R. Studer. 1999. Ontologies as Conceptual Models for XML Documents. In *Proceedings of the 12th International Workshop on Knowledge Acquisition, Modelling and Mangement (KAW'99), Banff, Canada, October*.

M. Erdmann, M. Maedche, S. Staab, and S. Decker. 2000. Ontologies in RDF(S). Technical Report 401, Institute AIFB, Karlsruhe University.

J. Heflin, J. Hendler, and S. Luke. 1999. Applying Ontology to the Web: A Case Study. In *Proceedings of the International Work-Conference on Artificial and Natural Neural Networks, IWANN'99*.

I. Horrocks et.al. 2000. The ontology interchange language oil: The grease between ontologies. Technical report, Dep. of Computer Science, Univ. of Manchester, UK/ Vrije Universiteit Amsterdam, NL/ AIdministrator, Nederland B.V./ AIFB, Univ. of Karlsruhe, DE. http://www.cs.vu.nl/˜dieter/oil/.

R. Hudson. 1990. *English Word Grammar*. Basil Blackwell, Oxford.

Michael Kifer, Georg Lausen, and James Wu. 1995. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42.

D. E. Knuth. 1984. Literate programming. *The Computer Journal*, 27:97–111.

O. Lassila and R. Swick. 1999. Resource description framework (RDF) model and syntax specification. Technical report, W3C. W3C Recommendation. http://www.w3.org/TR/REC-rdf-syntax.

S. Luke, L. Spector, D. Rager, and J. Hendler. 1997. Ontology-based Web Agents. In *Proceedings of First International Conference on Autonomous Agents, LNCS*.

A. Maedche and S. Staab. 2000a. Discovering conceptual relations from text. In *Proceedings of ECAI-2000*. IOS Press, Amsterdam.

A. Maedche and S. Staab. 2000b. Semi-automatic engineering of ontologies from text. In *Proceedings of the 12th Internal Conference on Software and Knowledge Engineering. Chicago, USA, July, 5-7, 2000*. KSI.

G. Neumann, R. Backofen, J. Baur, M. Becker, and C. Braun. 1997. An information extraction core system for real world german text processing. In *In Proceedings of ANLP-97*, pages 208–215, Washington, USA.

J. Pustejovsky, B. Boguraev, M. Verhagen, P. Buitelaar, and M. Johnston. 1997. Semantic indexing and typed hyperlinking. In *Proceedings of AAAI Spring Symposium, NLP for WWW*.

H.-P. Schnurr and S. Staab. 2000. A proactive inferencing agent for desk support. In *Proceedings of the AAAI Symposium on Bringing Knowledge to Business Processes*, Stanford, CA, USA. AAAI Technical Report, Menlo Park.

S. Staab and A. Maedche. 2000. Ontology engineering beyond the modeling of concepts and relations. In *Proceedings of the ECAI'2000 Workshop on Application of Ontologies and Problem-Solving Methods*.

S. Staab, C. Braun, A. Düsterhöft, A. Heuer, M. Klettke, S. Melzig, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, and B. Wrenger. 1999. GETESS — searching the web exploiting german texts. In *Proceedings of the 3rd Workshop on Cooperative Information Agents*, LNCS, Berlin. Springer.

S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, R. Studer, and Y. Sure. 2000. Semantic Community Web Portals. In *Proceedings of the 9th World Wide Web Conference (WWW-9), Amsterdam, Netherlands*.

F. van Harmelen and D. Fensel. 1999. Practical Knowledge Representation for the Web. In *Proceedings of the IJCAI Workshop on Intelligent Information Integration*.