Church

Second Annual Workshop on Very Large Corpora (WVLC2)

Thursday, 4 August 1994

Kyoto International Community House

Kyoto, Japan

Second Annual Workshop on Very Large Corpora (WVLC2)

Program and Proceedings

Thursday, 4 August 1994

Kyoto International Community House

Kyoto, Japan

Workshop Program

Corpora

09:30 McKelvie, David; Thompson, Henry S.;
TEI Conformant Structural Markup of a Trilingual Corpus in the ECI Multilingual Corpus 1

09:55 Yarowsky, David;
A Comparison of Corpus-based Techniques for Restoring Accents in Spanish and French Text

10:20 Uramoto, Naohiko;
Extracting Disambiguated Thesaurus from Parallel Dictionary Definitions

10:45 Kita, Kenji; Omoto, Takashi; Yano, Yoneo; Kato, Yasuhiko;
Application of Corpora in Second Language Learning - The Problem of Collocational Knowledge Acquisition

11:10 break

Alignment

11:35 Grishman, Ralph
Iterative Alignment of Syntactic Structures for a Bilingual Corpus

12:00 Fung, Pascale; Wu, Dekai;
Statistical Augmentation of a Chinese Machine-Readable Dictionary

12:25 break for lunch

14:00 Invited Speaker: Martin Kay

15:00 break

Information Retrieval and Matching

15:25 Fujii, Hideo; Croft, Bruce;
Comparing the Retrieval Performance of English and Japanese Text Databases

15:50 Merkel, Magnus; Nilsson, Bernt; Ahrenberg, Lars;
A Phrase-Retrieval System Based on Recurrence

16:15 Sekine, Satoshi;
Automatic Sublanguage Identification for a New Text

16:40 Umemura, Kyoji;
String Comparison Based on Substring Equations

Table of Contents

Authors' Index

TEI-Conformant Structural Markup of a Trilingual Parallel Corpus in the ECI Multilingual Corpus 1

David McKelvie & Henry S. Thompson

Human Communication Research Centre, University of Edinburgh
2 Buccleuch Place, Edinburgh, Scotland. <eucorp@cogsci.ed.ac.uk>

Abstract

In this paper we provide an overview of the ACL European Corpus Initiative (ECI) Multilingual Corpus 1 (ECI/MC1). In particular, we look at one particular subcorpus in the ECI/MC1, the trilingual corpus of International Labour Organisation reports, and discuss the problems involved in TEI-compliant structural markup and preliminary alignment of this large corpus. We discuss gross structural alignment down to the level of text paragraphs. We see this as a necessary first step in corpus preparation before detailed (possibly automatic) alignment of texts is possible.

We try and generalise our experience with this corpus to illustrate the process of preliminary markup of large corpora which in their raw state can be in an arbitrary format (eg printers tapes, proprietary word-processor format); noisy (not fully parallel, with structure obscured by spelling mistakes); full of poorly documented formatting instructions; and whose structure is present but anything but explicit. We illustrate these points by reference to other parallel subcorpora of ECI/MC1. We attempt to define some guidelines for the development of corpus annotation toolkits which would aid this kind of structural preparation of large corpora.

1. Overview of the ECI Corpus

1.1. Brief History and Acknowledgements

The ECI arose as a result of a concern shared by a number of European researchers in computational linguistics that waiting for fully funded support for collection and distribution of non-English corpus material would mean waiting too long. This concern crystalised into action, modelled on the Association for Computational Linguistics (ACL) Data Collection Initiative, following a meeting in Pisa sponsored by the Network for European Reference Corpora (NERC) in 1992. The original call for contributions to the ECI described it as follows:

> The European Corpus Initiative was founded to oversee the acquisition and preparation of a large multi-lingual corpus to be made available in digital form for scientific research

at cost and without royalties. We believe that widespread easy access to such material would be a great stimulus to scientific research and technology development as regards language and language technology. We support existing and projected national and international efforts to carefully design, collect and publish large-scale multi-lingual written and spoken corpora, but also believe it will be some time before the scientific and material resources necessary to bring these projects to fruition will be found. In the interim, a small and rapid effort to collect and distribute existing material can serve to show the way. No amount of abstract argument as to the value of corpus material is as powerful as the experience of actually having access to some in one's laboratory. We aim to make that experience possible very soon, at a very low cost.

The ECI is carrying out the first phase of this activity on a purely voluntary basis, under the guidance of an ad-hoc steering committee.

The majority of the work of collecting materials and permissions and converting them into a consistent format has been done at the Human Communication Research Centre, University of Edinburgh and at ISSCO, University of Geneva, under the overall supervision of Henry S. Thompson and Susan Armstrong, respectively. In addition to the infrastructure support provided by these institutions, modest financial contributions were provided by the European Network for Language and Speech (ELSNET), the LDC (University of Pennsylvania) and NERC. The ACL and the Language Technology Group of HCRC provided loans.

1.2. How to Acquire the ECI/MC1 CD-ROM

The CD-ROM is available in the US from the Linguistic Data Consortium (LDC), for members of the LDC or those making a bulk purchase, and otherwise from ELSNET, 2 Buccleuch Place, Edinburgh EH8 9LW, SCOTLAND. The cost from ELSNET is £20 plus postage, handling and tax where applicable, on signature of the necessary User Licence Agreement. Information about ordering the corpus can be had from Leeann Jackson-Eve <leeann@cogsci.ed.ac.uk>. Further information about the contents or markup in the corpus can be obtained from the authors of this paper at <eucorp@cogsci.ed.ac.uk>.

1.3. Overview of the contents of the ECI/MC1 corpus

The ECI/MC1 corpus contains almost 100 million words in 27 (mainly European) languages. It is now available on CDROM for research purposes at a low price. It consists of 48 opportunistically collected sub-corpora marked up using TEI P2 conformant SGML (to varying levels of detail). The sub-corpora vary considerably in size, the larger corpora include:

GER03 German Newspaper texts from the Frankfurter Rundschau
July 1992 - March 1993
Provided by Universitaet Gesamthochschule Paderborn Germany
Approximately 34 million words.

FRE01 French Newspaper texts from Le Monde
September, October 1989, and January 1990
Provided by LIMSI CNRS, France
Approximately 4.1 million words

DUT02 Extracts from the Leiden Corpus of Dutch
(newspapers, transcribed speech, etc)
Provided by Instituut voor Nederlandse Lexicologie, Leiden, Holland
Approximately 5.5 million words

MUL05 International Labour Organisation reports of the
Committee on Freedom of Association 1984-1989.
Parallel texts in English, French and Spanish
Approximately 1.7 million words per language

The corpus contains a number of parallel multilingual corpora. This paper will concentrate on the markup of one of these, the trilingual MUL05 corpus of International Labour Organisation reports.

2. Markup of the ILO corpus

2.1. Conversion of data into standard ASCII text files

The ILO corpus originally came to us from the International Labour Organisation in the form of a backup tape containing word processor files. Reading the tapes and converting their format into standard UNIX format text files was a non-trivial operation, which was undertaken by Dominique Petitpierre (ISSCO) and David Graff (LDC). The details of this step are unimportant for the purposes of the present paper, except to note that this initial stage of processing is often necessary and is often made difficult by lack of easily available documentation and requires specially written software.

The corpus files originally used the Wang WP ASCII World Languages Character Set. This character set includes normal characters, underlined versions of these characters, plus some control characters to control rendition features such as centering lines, indentation, sub- and super-script characters. Following information obtained from Wang, ISSCO and LDC converted the files to use the ISO-LATIN-1 character set with SGML-style markup for the rendition features.

2.2. Conversion of control characters into markup

The texts contain some markup which describes the physical shape and position of the characters. They have been converted into SGML rendition attributes attached either to a structural division ( ) or to a <hi> ... </hi> section where necessary. If more than one rendition is applicable to a section of text, then the rendition attribute of the section is formed by concatenating the different rendition values separated by "." eg

<hi rend=ul.centr> Means underlined and centered text

There are difficulties in converting a command based markup scheme into a structural markup scheme. In a command based scheme control characters/sequences are used to change the state of some formatting machine in a sequential fashion. Thus the sequence:

[bold on] aaa [italic on] [bold off] bbb [italic off]

is a legal sequence of instructions to the word processor. Such sequences were a common occurence in this corpus. Converting this to SGML markup in a straightforward way, leads to the badly nested structure:

<bold> aaa <italic> </bold> bbb </italic>

since SGML describes the structure of text in a hierarchical fashion, (We assume here that we prefer not to use SGML processing instructions to mark these font changes). This is a common problem when converting such WP texts into SGML. Corpus annotation toolkits should provide tools for converting such sequences into valid SGML markup.

Again, without a detailed understanding of the word-processor's operations it is not always straightforward to change its control codes into structural markup e.g. in the sequence (line breaks as in the original)

A line of text
[underline on]
[underline off]

the underlining would appear to have no effect. In fact, it became clear that the underlining was active for the full defined width of the line, thus underlining the previous line of text. Thus the correct SGML markup was:

<hi rend=ul>A line of text</hi>

Another nice example of the difficulties of attempting SGML markup which captures the semantics of the texts is the following: Some section titles had every second character underlined, originally this would have the appearance of text with a broken line underneath it. By the time we had SGML marked this up, it looked like:

<hi rend=ul>A</hi> <hi rend=ul>s</hi>e<hi rend=ul>c</hi>
t<hi rend=ul>i</hi>o<hi rend=ul>n</hi> <hi rend=ul>t</hi>i<hi rend=ul>t</hi>
l<hi rend=ul>e</hi>

which is totally unreadable, fails to capture the semantics of the markup in a clear and descriptive way and makes searching for words in the text difficult. Instead we introduced and documented a new value of the rendition attribute and recoded this as:

<hi rend=ul2>
A section title
</hi>

Similar cases occur in newspaper texts where the first letter of the first word of an article is in bold font and the rest of the word in normal font. In this case we have placed the whole word in a <hi rend=first.letter.bold> element.

2.3. Text markup invariants

We take the approach that although TEI markup is the correct technique for text annotation, not all users of our corpus will use SGML to access this data. We thus tried to keep text and markup separate from each other. The bulk of the data provided observes what we call the Text/Markup Invariant: Every line in a data file is either all text or all markup, and a line is a markup line if and only if it begins with a left angle bracket (<). This makes restricting your processing to 'plain text' very easy – just look only at lines which begin with some character other than <.

2.4. Determining the logical structure of the corpus

The corpus came to HCRC from ISSCO in the form of 292 files named 'doc.<NNN>.iso'. This was the physical form of the corpus which appears to conform with the original division of the corpus into Wang word-processor files.

In contrast to the above, the logical structure of the corpus can be expressed as:

ILO → CORPUS(eng) CORPUS(fre) CORPUS(spa)
CORPUS(LANG) → VOLUME(LANG)+
VOLUME(LANG) → ISSUE(LANG)+ SPECIAL-SUPPLEMENT(LANG)*
ISSUE(LANG) → CONTENTS(LANG) REPORT(LANG)+

That is, the ILO corpus is made of three language-specific corpora, in English, French and Spanish. Each language-specific corpus is made up of a number of volumes each of which contains all of the publications in that language in a particular year. Each Volume consists of a number of issues (normally three) and a number of special supplements, each of which consists of a single publication. Each ISSUE contains a table of contents and a number of REPORTS. Special Supplements do not contain reports within them. Each ISSUE has a table of contents, covering all the reports in the issue.

We decided to take the single-language ISSUE as the basis for our re-organisation of the corpus, that is, all the material from a single issue would be placed together into a single computer file. The first stage of the reorganisation of the corpus was to collect all of the files which contained information from a single one-language issue into a single computer file. Fortunately, the information needed to do this was contained in header blocks at the front of each WP file. In general, the re-organisation of a corpus into meaningful pieces is not a straightforward operation, and requires an understanding of the contents of the data.

Since the division of the documents into original files cut across the hierarchical structure of the documents, in the new single issue files the locations of the original file breaks was marked with the SGML <milestone> tag as follows:

<milestone unit=file n="Original File Name">

Each of the 'doc.<NNN>.iso' files had a header block at the start. These header blocks were removed from the new versions of the files since the information they contained was made explicit in the new structure.

2.5. Markup of structural text divisions

The ILO documents were highly structured, the difficulty was in capturing this structure in SGML. The lowest clearly marked level of structure in the documents was the numbered paragraph. These are consecutively numbered from 1 in each report or appendix. They have been marked-up as

<divN n={original number} type=ILOpara>

The value of N reflects where the ILO paragraph is in the document structure and can vary, it is typically 6.

Above this basis of ILO paragraphs, higher level sections were constructed, partly by inspection of the texts and partly by comparison with the table of contents. These are marked up using

<divN type={SectionType}>

where N varies depending on the depth of the structure nesting and SectionType describes the semantics of the division e.g. "Report", "Case", "CaseRecommendation" etc. Fortunately, most section titles were marked syntactically, for example with bold fonts, and the reports follow a fairly constant structure. The biggest difficulty was in determining the nesting structure of the divisions and the types of the divisions.

We decided to use numbered divisions <div1> etc rather than un-numbered divisions e.g. because end tags can be added automatically, and navigating around the files is made easier. Also, in this corpus, the type of a division did not always correlate with its level of nesting.

Since the higher level divisions were not explicitly marked in the text, we made an effort to determine the type of a division from its title. This was only possible due to the stereotyped nature of the reports. For example, each case report had a section giving the recommendations of the committee. However these were not so easy to find. The final search pattern which we used to find such titles was:

/The Committee's recommen?dations?/
/The recommen?dations? of the Committee/
/Recommandations? +du +comité/
/Recomendaci(o|ó)n(es)? del Comit(é|e)/

(? means the previous item is optional, + means the previous item can be repeated more than once, (a|b) means a or b)

As can be seen, it would have been difficult to determine these patterns without an exhaustive search of the corpus. In fact they were arrived at by a process of iterative refinement.

Occasionally divisions have been introduced below the ILO paragraph level. Normally however, below the ILO paragraph level we have only marked sections separated by a blank line with <p>

... </p>. These normally contain running text, but were also used for items in lists and headings in tables etc.

A fairly complex perl script was written to process the files. The new files were then checked and edited by hand.

Another complication was the presence of footnotes and references to footnotes. Footnotes and references to them were marked in a number of slightly different ways in the text. For example, some were marked by a special control sequence, others were marked simply by superscript numbers in running text. These different ways had to be found and their difference in meaning (if any) discovered. This was complicated by the existence of footnotes running over several pages and even footnotes inside other footnotes. In the case of superscript numbers, all but two such occurrences were in fact footnote references. The other two were of the form "m^2" and meant square metres! Again without careful checking of all occurrences, such things will go unnoticed. Finally we added SGML IDs to all notes and cross-referenced them to the references to the footnotes. By doing so, we uncovered some more notes which due to slight differences in their syntax had been missed.

2.6. Cleanup of texts

A fairly common error in these files was the occurrence of the letters "l" and "O" in contexts where it was clear that they should have been the numbers "1" and "0" respectively. This is also a common error in files which have been OCR scanned. It proved easy enough to write a simple pattern which matched such letters in numeric contexts and convert them to numbers. However, before this error was discovered, it caused some other patterns looking for e.g. paragraph numbers, to fail and hence cause further errors in the markup.

Line internal hyphenation (eg a word followed by a hyphen and a space) have been quietly edited either to a complete word or to a hyphenated word where appropriate. I.e. except where it is clearly part of the significant layout of say headers and/or tables or when it part of a multi-word compound e.g. 'two- or three-weekly'. Line final hyphenation has been left as is.

There is still a considerable amount of further markup which could be applied to this corpus. For example, we made no attempt to markup tables or lists in any special way. No dates or proper names, which are an important part of the information content of this corpus, were marked as such.

2.7. Alignment of structure across three languages

Once the individual files had been marked-up using TEI structural markup e.g. major textual divisions, titles and paragraphs, the different language versions were compared with each other.

Using simple perl scripts and UNIX tools the different language versions were compared and were edited to ensure that the structure of each version was the same. As far as possible, each language version of a document has the same SGML structure as far as <divN> elements are concerned. This means that if there is a <divN> in (say) the English version of a report, then there will be a similar <divN> in the French and Spanish versions, containing equivalent text. It is not guaranteed that the language versions are parallel down to the SGML <p> level. In most cases a <p> in one language corresponds to a <p> in the other languages, but not always.

This process was complicated by the following facts:

- Equivalent sections were in different positions in different versions, in particular appendices and tables were sometimes placed inline and sometimes placed at the end of a file.
- Sometimes there were missing sections in one of the languages, or extra notes or draft sections existed in one language.
- Sometimes previous markup processing had missed a section or a section title in one version. For example, normally section titles are separated from their text by blank lines. In the Spanish texts, some paragraph titles were included in the paragraph text and not specially marked. This was not noticed until we compared them with the other language versions.

3. Conclusions

The markup of this corpus involved a great deal of inspection of the data, looking for common patterns, determining their meaning or lack of meaning and devising TEI conformant markup which captured this meaning. Then pattern-matching programs which could convert the existing data into the desired new form had to be written. These transformations often required an iterative process of making edits, checking the results and modifying the edits in order to capture the full range of variability in the texts.

For example, in Spanish, ordinal numbers are often followed by a superscript *o* , however sometimes this is represented as the character º and sometimes by the sequence <superscript>o</superscript>, i.e. different typists use different methods to achieve the same end.

Until all producers of large documents produce their files in a common SGML format, there will be a role for this kind of document markup. It is certainly of interest to create tools which speed up this process. However, it is not as simple as writing a pattern matching program and running it over the data.

In the first case, it is difficult to write patterns which match all and only the desired data. For large complex texts it is necessary to assume that any pattern will need refinement and correction until it correctly matches ones intuitive understanding and the nature of the text. One can either impose structure on the texts, e.g. *define* a number to be a sequence of digits and stick to this definition. Or one must be prepared to allow the actual data to redefine or refine ones definition of what a number is e.g. allow I for 1 mis-typings etc.

Secondly, SGML markup is inherently recursive - finite state languages (i.e. most pattern matching languages) - find this difficult or impossible to deal with in full generality.

Finally, although we started with a clear definition of the markup that we wanted to add and how we would do it, we found that after 600 megabytes of data that our ideas about markup had considerably changed in order to model the data. With every such change one has the problem of checking that previously marked up subcorpora are still conformant.

Our suggestions would be:

- There is a role for imposing textual invariants on the data, for example removing all line final white space or placing all markup on separate lines. Environments which make it easy to define such invariants and which help in checking/enforcing them are needed.
- There is likewise great scope for having a library of small software tools which handle common tasks e.g. hyphenation or reordering font changes to make valid SGML.
- However there will always be a need for a text processing environment where it is easy to write programs to check and alter texts. Each new corpus is different. It would be a great help to the corpus annotator if such an environment could look after the common features of text processing, such as backup of old versions and checking planned edits for 'dangerous' operations.
- The ECI/MC1 corpus was annotated using mainly perl, emacs and various UNIX tools. This provided a reasonable text processing environment, but it could be improved. Describing how it could be improved soon descends to a long list of disconnected facts, for example, that "wc" (UNIX tool for counting words) does not work correctly on files containing 8-bit characters or that (unless one tweaks an obscure variable) Emacs will change the case of letters in edits in some contexts. It tries to be helpful, but sometimes gets it horribly wrong. Designing a text processing toolkit which is powerful, intelligent and transparently clear in its operation is needed, but almost certainly a pipe dream.

- Given this, we need better tools for searching and scanning data to find anomalous patterns in the data.

References

[1] "Practical SGML", Eric van Herwijnen, Kluwer Academic Publishers, 1990.

[2] "The SGML Handbook", Charles F. Goldfarb, Clarendon Press, 1990.

[3] "Programming perl", Larry Wall and Randal L. Schwartz, O'Reilly and Associates, Inc. , 1991.

[4] "Emacs users guide", Online documentation available with emacs.

[5] "Tutorial on Text Corpora", Mark Liberman and Mitch Marcus, ACL '92.

[6] "Guidelines for Electronic Text Encoding and Interchange", C.M.Sperberg-McQueen and Lou Burnard (eds), Pre-publication P2 drafts, Text Encoding Initiative, 1993. [Available from the Listserv list TEI-L in electronic form]

A Comparison of Corpus-based Techniques for Restoring Accents in Spanish and French Text

David Yarowsky*
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
yarowsky@unagi.cis.upenn.edu

Abstract

This paper will explore and compare three corpus-based techniques for lexical ambiguity resolution, focusing on the problem of restoring missing accents to Spanish and French text. Many of the ambiguities created by missing accents are differences in part of speech: hence one of the methods considered is an N-gram tagger using Viterbi decoding, such as is found in stochastic part-of-speech taggers. A second technique, Bayesian classification, has been successfully applied to word-sense disambiguation and is well suited for some of the semantic ambiguities which arise from missing accents. The third approach, based on decision lists, combines the strengths of the two other methods, incorporating both local syntactic patterns and more distant collocational evidence, and outperforms them both. The problem of accent restoration is particularly well suited for demonstrating and testing the capabilities of the given algorithms because it requires the resolution of both semantic and syntactic ambiguity, and offers an objective ground truth for automatic evaluation. It is also a practical problem with immediate application.

1 PROBLEM DESCRIPTION

Accent restoration is closely related to several lexical disambiguation problems. It involves aspects of both word-sense disambiguation and part-of-speech tagging. While not as widely cited as these other tasks, it nonetheless offers considerable benefits as a case study, and is particularly useful for evaluating and comparing the disambiguation algorithms considered here. Specifically:

- It requires the resolution of both syntactic and semantic ambiguities, and is representative of many of the issues that arise in several important types of lexical ambiguity resolution.
- Unlike many ambiguity resolution tasks which depend on human annotations or judgements for evaluation, this problem supports fully automatic evaluation and an innate, plentiful and objective ground truth – text with accents may be artificially stripped, leaving accentless text for testing purposes with a known gold standard for evaluation.
- The problem has immediate and practical application, both as a stand-alone product and a front-end component to multilingual NLP systems. There is also a large potential commercial market in its use in grammar and spelling correctors, and in aids for inserting the proper diacritics automatically when one types. Such a tool would be particularly useful for typing

*This research was supported by an NDSEG Fellowship, ARPA grant N00014-90-J-1863 and ARO grant DAAL 03-89-C0031 PRI. The author is also affiliated with the Linguistics Research Department of AT&T Bell Laboratories, and greatly appreciates the use of its resources in support of this work. He would like to thank Jason Eisner, Libby Levison, Mark Liberman, Mitch Marcus, Joseph Rosenzweig and Mark Zeren for their helpful feedback.

Spanish or French on Anglo-centric computer keyboards, where entering accents and other diacritic marks every few keystrokes can be laborious.

Thus while accent restoration may not be be the prototypical member of the class of lexical-ambiguity resolution problems, it is an especially useful one for describing, evaluating, and comparing proposed solutions to this class of problems.

Accent ambiguities arise routinely under a number of circumstances in Spanish and French[1]. It is traditional in both languages for diacritics to be omitted from capitalized letters. This is particularly a problem in all-capitalized text such as headlines. Accents in on-line text may also be systematically stripped by many computational processes which are not 8-bit clean (such as some e-mail transmissions), and may be frequently omitted by Spanish and French typists in informal computer correspondence.

Limited space precludes a full discussion of the range of accent pattern ambiguities encountered; see [Yarowsky, 1994] for more detail. Discussion here will focus on the following types of ambiguity in Spanish: The most common ambiguity is between the endings *-o* and *-ó*, as in *marco* vs. *marcó*. This is typically both a verb-tense and part-of-speech ambiguity. The second most common general ambiguity is between the past-subjunctive and future tenses of nearly all -ar verbs (eg: *terminara* vs. *terminará*), both of which are 3rd person singular forms. This is a particularly challenging class and is not readily amenable to traditional part-of-speech tagging algorithms such as local trigram-based taggers. Other ambiguities include function words (*mi* vs. *mí*) and purely semantic ambiguities such as *secretaria* (secretary) vs. *secretaría* (secretariat). The distribution of ambiguity types in French is similar, including the frequent part-of-speech and/or tense ambiguity between *-e* and *-é* endings, and numerous semantic ambiguities such as *traité/traite* (treaty/draft).

2 COMPARISON OF ALGORITHMS

This section will describe the application of 4 algorithms to the problem of accent restoration, outlining the details of the implementations and the performance achieved.

2.1 Method 1: Baseline

The vast majority of tokens in Spanish and French exhibit only one accent pattern. And in cases where there is ambiguity, one pattern is typically dominant. Thus one can achieve surprisingly good performance by using only the most common accent pattern for each token. This baseline approach is the standard by which all other techniques will be measured.

Initial corpus analysis will yield accent pattern distributions such as the following (for French):

De-accented Form	Accent Pattern	%	Number
cesse	cesse	53%	669
	cessé	47%	593
cout	coût	100%	330
couta	coûta	100%	41
coute	coûté	53%	107
	coûte	47%	96
cote	côté	69%	2645
	côte	28%	1040
	cote	3%	99
	coté	<1%	15
cotiere	côtière	100%	296

[1]For brevity, the term *accent* will typically refer to the general class of accents and other diacritics, including ê,è,é,õ etc. The term *accent restoration* should more accurately be called *diacritic restoration*.

Measured over all tokens, the baseline approach achieves 98.7% mean accuracy for Spanish and 97.6% mean accuracy for French. A breakdown of the baseline performance on the words used in the comparative study is given in Table 1, under the column labeled "BaseL". While this base performance may seem high, it still produces an error every 40-75 words in text. More importantly, the cases that it misses are precisely those where accents resolve an ambiguity, and thus the most important to handle correctly. Some attempt to resolve these ambiguities is clearly warranted.

2.2 Method 2: N-Gram Tagger

Since most of the ambiguities due to missing accents correspond to differences in parts-of-speech, it is natural to consider the algorithm most commonly applied to the problem of part-of-speech tagging, namely the Markov or N-Gram tagger. This approach was first widely publicized in [Church, 1988] and has become the standard in the field[2].

It is not necessary, however, to train a full part-of-speech tagger for Spanish and French to restore accents. Many part-of-speech distinctions have no direct bearing on choice of accent pattern. It may be advantageous to build an n-gram tagger which focuses only on the distinctions necessary to resolve the major accent ambiguities (e.g., *-o/-ó*, *-ara/-ará*, *-aran/-arán* in Spanish), ideally using only information available in an unannotated corpus.

One natural approach in morphologically rich languages such as Spanish and French is to build a model not of part-of-speech sequences, but of *suffix* sequences. It would be desirable if the patterns of suffixes and a small set of function words in nearby context were adequate to disambiguate ambiguous forms. The only linguistic knowledge that would be necessary then is a list of suffixes and function words in the language. Given this, it is straight-forward to create a training set like the following, with words annotated with their suffix or function word label, and the word form then stripped of any accents:

la posición anunció oficialmente que $\Longrightarrow$
`... la/LA posicion/-IÓN anuncio/-Ó oficialmente/-MENTE que/QUE ...`

cambiar el anuncio utilizado ... a mí $\Longrightarrow$
`... cambiar/-AR el/EL anuncio/-O utilizado/-ADO ... a/A mi/MÍ`

Using an N-Gram tagger trained on such data, one can recover the most probable suffix annotations for a sequence of de-accented text. Given that a word's accent pattern is almost always unambiguous given a suffix and can be described in a table such as in Method 1 above, the disambiguation process is a straightforward application of the channel model. The actual algorithm used in this experiment is described in [Rabiner 1989] and [Paul 1990]. The B-matrix emit probabilities are defined as $B[TAG, deaccented\_token] \equiv p(deaccented\_token|TAG)$, with transition probabilities defined analogously. The most probable tag sequence for new test data is recovered using a standard Viterbi decoder, implemented from the description in [Rabiner 1989].

The particular use of function-word and suffix sequences has several advantages, the foremost being that no large-scale lexical resources or annotated corpora are required; raw (accented) text is used for training. It is most viable in morphologically rich languages, and may be extended naturally to a full part-of-speech tagger through EM iteration.

The approach exhibits several weaknesses, however. The first is that there are many suffixes in Romance languages, yielding very large matrices and sparse data. It would clearly be desirable to recognize that the suffixes *-aba* and *-ía* both represent the same 1st/3rd person singular imperfect tense (just for different conjugation paradigms) and are functionally equivalent. This and further clustering can be accomplished either manually or by empirical induction. However, a greater

[2]Note that because the techniques in Methods 2 and 3 have been so thoroughly presented elsewhere, they will be covered somewhat briefly here to allow more space to be devoted to the new Method 4.

problem is the noise introduced when several parts of speech have the same suffix. For example, some nouns may also end in *-aba* and *-ía*, although these are primarily imperfect tense markers. This noise may be tolerable given the relatively low entropy of $p(part\_of\_speech|suffix)$ in Romance languages, but it is apparent that improvement could be achieved using existing dictionary resources to distinguish such cases.

This basic suffix model was implemented with all words assigned automatically to the longest match in a list of 54 suffixes and 40 common function words, with the residual labeled with one of 6 simple classes including punctuation and number. Performance is presented in Table 1 (SUFFIX).

Another variant of Method 2 that was tested here is to use additional dictionary resources in the spirit of [Merialdo, 1990], specifically with the Collins Spanish-English Dictionary and the Liberman-Tzoukermann morphological transducer [1990] used for extrapolation to inflected forms. In this study, the tags are traditional parts of speech (e.g. ADV, ADJ, SPRON, PASTPART), plus individual tags for important function words (e.g. QUE, ...). Suffixes involved in accent ambiguities (-ARA, -ARÁ, -ARAN, -ARÁN, etc.) are given their own tag to allow for specialized context modelling for each of these cases.

For the part-of-speech tags, it is assumed that a word may be tagged with each entry listed in the dictionary for that word with equal probability, with the residual receiving a small epsilon probability. For the suffix tags, the true probability distributions can be extracted from the training corpus. Thus the relative probability of the de-accented *anuncio* being a past-tense verb (*anunció*) or noun (*anuncio*) is directly measured and exploited in classification.

Performance of this specific approach is given in Table 1 (P.O.S.), broken down by general ambiguity type[3]. These results are based on a tagset of 61 parts of speech. Although this variant of Method 2 makes use of dictionary knowledge not available in the suffixes themselves, it uses a smaller tagset (including fewer function words) and makes fewer lexical distinctions, which may explain why the suffix-only method sometimes outperforms it.

TABLE 1: N-Gram Tagger Performance

Pattern 1	Pattern 2	SUFFIX	P.O.S.	BaseL	N
anuncio	anunció	97.4%	95.8%	57%	9459
registro	registró	97.0%	97.0%	60%	2596
marco	marcó	97.8%	97.5%	52%	2069
completo	completó	92.6%	85.2%	54%	1701
retiro	retiró	97.0%	97.3%	56%	3713
duro	duró	90.3%	93.7%	52%	1466
paso	pasó	88.0%	93.9%	50%	6383
regalo	regaló	89.5%	89.5%	56%	280
terminara	terminará	60.0%	65.7%	59%	218
llegara	llegará	68.9%	67.6%	64%	860
esta	está	88.7%	85.8%	61%	14140
mi	mí	90.0%	94.1%	82%	1221
secretaria	secretaría	52.3%	52.3%	52%	1065

The reader will notice considerable opportunity for further improvement in this approach. A hand-tagged corpus could be used for better initial probability estimation, and the EM algorithm could be used to refine B-Matrix probabilities iteratively [Merialdo, 1990]. However, the goal of this study was not to produce a full part-of-speech tagger, but to improve ambiguity resolution in accent

[3]The words used in this comparative study are a random selection from the most problematic cases of each ambiguity type – those exhibiting the largest absolute number of the non-majority accent patterns. Collectively they are representative of the most common potential sources of error. The training and test sets were independent in all cases, and the examples were extracted from the Spanish AP Newswire (1991-1993, 49 million words). These same words were also used to test all the other methods in this comparative study.

restoration. A cost-benefit analysis could help determine whether additional resources are worth devoting to this approach.

This method has several fundamental limitations for the task of accent restoration. First, it is not adequately lexicalized. For example, for the *-ara/-ará* subjunctive/future distinction, the presence of temporal words (days of the week, months, events, etc.) are highly significant, and for other tense distinctions specific lexical associations are important. One could add additional word classes, but there are many more useful distinctions than can be adequately accommodated given the algorithm's time and space complexity bounds. More intractably, however, many of the necessary tense distinctions are sensitive to mid-to-long distance word associations (such as the temporal indicators) that simply cannot be captured with an n-gram model, for any reasonable size of *n*. And finally, the approach does not address the cases that arise when a token has multiple accented forms with the same part of speech. It would appear that further progress can best be made by developing more lexicalized and longer-distance models of context.

2.3 Method 3: Bayesian Classifier

Bayesian classifiers are particularly well suited for handling highly lexicalized and longer-distance models of context, two of the central weaknesses of the previous approach. They have been employed successfully in word-sense disambiguation [Gale, Church and Yarowsky, 1992B], authorship identification [Mosteller and Wallace, 1964] and person-place classification of proper nouns [Gale, Church and Yarowsky, 1992].

The basic technique employed is to treat a window of words surrounding each ambiguous word as a document, and ask if there are any measurable differences in the distribution of words found in the contexts surrounding one of its accent patterns relative to the other. For example, when distinguishing the accent patterns *terminara* (subjunctive) from *terminará* (future), one would tend to find that the token *domingo* (Sunday) occurs much more frequently in the context of the latter than the former, while certain subjunctive marking phrases occur in an inverse distribution. Considering only one of these words in context, we can estimate the probability that the context belongs to one accent pattern relative to another by the likelihood ratio:

$$\frac{p(token\_in\_context|accent\_pattern_1)}{p(token\_in\_context|accent\_pattern_2)}$$

Making the simplifying assumption that all tokens seen in the context of an ambiguous word provide *independent* evidence for classifying the accent pattern, we can combine the ratios in a product to yield an overall likelihood ratio that the ambiguous token has one accent pattern relative to another:

$$\prod_{token \text{ in } context} \frac{p(token_i|accent\_pattern_1)}{p(token_i|accent\_pattern_2)}$$

A primary variable here is the width of context considered. Two experiments were conducted, one examining a fairly wide context (± 20 words) and one examining a more localized context ($\pm 2-4$ words). The larger is similar to the width often employed in sense disambiguation, and is useful for modelling "semantic" or "topic" differences, while the smaller window is better suited for modelling more "syntactic" distinctions.

The following table provides an outline of the performance of Method 3 (Bayesian Classifiers), using context window sizes of ± 2, ± 4 and ± 20 words.

TABLE 2: Bayesian Classifier Performance

Pattern 1	Pattern 2	±2	±4	±20	BaseL
anuncio	anunció	85.5	88.4	74.7	57 %
registro	registró	87.1	81.8	77.0	60 %
marco	marcó	94.4	93.0	93.5	52 %
completo	completó	90.6	89.2	88.6	54 %
retiro	retiró	88.1	88.6	79.3	56 %
duro	duró	93.5	93.4	82.1	52 %
paso	pasó	88.2	86.5	76.4	50 %
regalo	regaló	84.7	80.4	75.9	56 %
terminara	terminará	79.2	83.5	82.8	59 %
llegara	llegará	65.6	72.2	62.9	64 %
esta	está	88.2	87.8	81.3	61 %
mi	mí	80.4	79.9	76.7	76 %
secretaria	secretaría	78.1	75.0	75.6	52 %

The Bayesian classifier has the advantage of not requiring special lexical resources or annotated corpora. It supports a highly lexicalized feature set and may capture long-distance dependencies. It can distinguish ambiguities within the same part of speech.

However, the major disadvantage of the "bag of words" Bayesian classifier approach is that it is difficult to model the occurrence of words in specific positions. Given the assumption of independence, it is also quite difficult to model *sequences* of nearby words; when the joint appearance of two or more words differ in their distribution from that expected from the product of their independent likelihood ratios. This independence assumption also makes the technique poorly suited for combining multiple non-independent sources of evidence, such as parts-of-speech, lemmas, word classes and individual inflected words all in the same context.

2.4 Method 4: Decision Lists

The limitations observed above are precisely what has motivated the development of Method 4, a hybrid approach using *decision lists*, combining the strengths of both Bayesian classifiers and N-gram taggers. This approach was derived from the formal model of decision lists presented in [Rivest, 1987]. However, feature conjuncts have been restricted to a much narrower complexity than allowed in the original model – namely to word and class trigrams. Early results presented in [Sproat, Hirschberg and Yarowsky, 1992] achieved 97% mean accuracy on the problem of homograph resolution in text-to-speech synthesis[4]. The current approach was proposed in [Yarowsky, 1994] and is described more fully there. Below is an outline of the algorithm:

Steps 1 & 2: Measure Accent Pattern Distributions and Collect Training Contexts

The algorithm begins by identifying the accent pattern ambiguities for a language. An accent distribution table is computed as described in Method 1 (Baseline). For each case of accent ambiguity identified, collect $\pm k$ words of context around all occurrences in the training corpus, label the concordance line with the observed accent pattern, and then strip the accents from the data. This will yield a training set such as the following:

[4]For the data set of 13 homographs used in this study, baseline correctness was 67%.

Pattern	Context
(1) côté	du laisser de *cote* faute de temps
(1) côté	appeler l' autre *cote* de l' atlantique
(1) côté	passe de notre *cote* de la frontiere
(2) côte	vivre sur notre *cote* ouest toujours verte
(2) côte	creer sur la *cote* du labrador des
(2) côte	travaillaient cote a *cote* , ils avaient

Step 3: Measure Collocational Distributions

The driving force behind this disambiguation algorithm is the uneven distribution of collocations[5] with respect to the ambiguous token being classified. The presence of certain collocations will indicate one accent pattern, while different collocations will tend to indicate another. The goal of this stage of the algorithm is to measure a large number of collocational distributions and select those which are most useful in identifying the accent pattern of the ambiguous word.

The following are the initial types of collocations considered:

- Word immediately to the left (-1 W)
- Word found in $\pm k$ word window[6] ($\pm$k W)
- Pair of words at offsets -2 and -1
- Pair of words at offsets -1 and +1
- Pair of words at offsets +1 and +2

For the two major accent patterns of the French noun *cote*, below is a small sample of these distributions for several types of collocations:

Position	Collocation	**côte**	**côté**
-1 W	du *cote*	0	536
	la *cote*	766	1
	un *cote*	0	216
	notre *cote*	10	70
+1 W	*cote* ouest	288	1
	cote est	174	3
	cote du	55	156
+1W,+2W	*cote* du gouvernement	0	62
-2W,-1W	cote a *cote*	23	0
$\pm$k W, $k = 20$	poisson (within $\pm$20 words)	20	0
$\pm$k W, $k = 20$	ports (within $\pm$20 words)	22	0
$\pm$k W, $k = 20$	opposition (within $\pm$20 words)	0	39

By themselves, such simple word associations have considerable discriminating power, and can successfully model gender constraints, etc. without these constraints being explicitly represented (or known). However, if additional resources such as a morphological analyzer are available, similar collocational patterns for linguistic features such as morphological root may be measured. This often yields more succinct and generalizable discriminators than achieved from a list of the observed inflected forms. The Tzoukermann/Liberman [1990] Spanish morphological analyzer was used here for this purpose. Similarly, distributional patterns for part-of-speech bigrams and trigrams were computed, using a relatively coarse level of analysis (such as NOUN, ADJECTIVE, SUBJECT-PRONOUN, ARTICLE, etc.) comparable to that used in Method 2. However, since the information

[5]The term *collocation* is used here in its broad sense, meaning words appearing adjacent to or near each other (literally, in the same location), and does not imply only idiomatic or non-compositional associations.

[6]The optimal value of k is sensitive to the type of ambiguity. Semantic or topic-based ambiguities warrant a larger window ($k \approx 20 - 50$), while more local syntactic ambiguities warrant a smaller window ($k \approx 3$ or 4)

was extracted from a dictionary and not from a part-of-speech-tagged corpus, no relative frequency distribution was available for words with multiple parts-of-speech. Such words were given a part-of-speech tag consisting of the union of the possibilities (eg ADJECTIVE-NOUN), as in Kupiec (1989). Thus sequences of pure part-of-speech tags were highly reliable, while the potential sources of noise were isolated and modeled separately. In addition, collocational statistics were measured for several word classes, such as WEEKDAY (={ *domingo*, *lunes*, *martes*, ... }) or MONTH, primarily focusing on time words because so many accent ambiguities involve tense distinctions.

To build a full part of speech tagger for Spanish would be quite costly (and require special tagged corpora). The current approach uses just the information available in dictionaries, exploiting only that which is useful for the accent restoration task. Were dictionaries not available, a productive approximation could have been made using the associational distributions of suffixes (such as *-aba*, *-aste*, *-amos*) which are often satisfactory indicators of part of speech in morphologically rich languages such as Spanish.

For the French experiments, no additional linguistic knowledge or lexical resources were used. The decision lists were trained solely on raw word associations without additional patterns based on part of speech, morphological analysis or word class. Hence the reported performance is representative of what may be achieved with a rapid, inexpensive implementation based strictly on the distributional properties of raw text.

The use of the word-class and part-of-speech data is illustrated below, with the example of distinguishing *terminara*/*terminará* (a subjunctive/future tense ambiguity):

Position	Collocation	**terminara**	**terminará**
-2P,-1P	PREPOSITION QUE *terminara*	31	0
-2W,-1W	de que *terminara*	15	0
-2W,-1W	para que *terminara*	14	0
-2P,-1P	NOUN QUE *terminara*	0	13
-2W,-1W	carrera que *terminara*	0	3
-2W,-1W	reunion que *terminara*	0	2
-2W,-1W	acuerdo que *terminara*	0	2
-1W	que *terminara*	42	37
$\pm$k C, $k = 20$	WEEKDAY (within $\pm$20 words)	0	23
$\pm$k W, $k = 20$	domingo (within $\pm$20 words)	0	10
$\pm$k W, $k = 20$	viernes (within $\pm$20 words)	0	4

Step 4: Sort by Log-Likelihood into Decision Lists

For each individual collocation, the following log-likelihood ratio was computed:

$$Abs(Log(\frac{p(Accent\_Pattern_1|Collocation_i)}{p(Accent\_Pattern_2|Collocation_i)}))$$

The collocations most strongly indicative of a particular pattern will have the largest log-likelihood. Sorting by this value will list the strongest and most reliable evidence first[7].

Evidence sorted in the above manner will yield a decision list like the following, highly abbreviated example[8]:

[7] Problems arise when an observed count is 0. Clearly the probability of seeing *côté* in the context of *poisson* is not 0, even though no such collocation was observed in the training data. Finding a more accurate probability estimate depends on several factors, including the size of the training sample, nature of the collocation (adjacent bigrams or wider context), our prior expectation about the similarity of contexts, and the amount of noise in the training data. Several smoothing methods have been explored here, including those discussed in [Gale et al., 1992B]. In one technique, all observed distributions with the same 0-denominator raw frequency ratio (such as 2/0) are taken collectively, the average agreement rate of these distributions with additional held-out training data is measured, and from this a more realistic estimate of the likelihood ratio (e.g. 1.8/0.2) is computed. However, in the simplest implementation, satisfactory results may be achieved by adding a small constant α to the numerator and denominator, where α is selected empirically to optimize classification performance. For this data, relatively small α (between 0.1 and 0.25) tended to be effective, while noisier training data warrant larger α.

[8] Entries marked with † are pruned in Step 5, below.

LogL	Evidence	Classification
8.28	PREPOSITION QUE *terminara*	⇒ terminara
†7.24	de que *terminara*	⇒ terminara
†7.14	para que *terminara*	⇒ terminara
6.87	y *terminara*	⇒ terminará
6.64	WEEKDAY (within ±20 words)	⇒ terminará
5.82	NOUN QUE *terminara*	⇒ terminará
†5.45	domingo (within ±20 words)	⇒ terminará

The resulting decision list is used to classify new examples by identifying the highest line in the list that matches the given context and returning the indicated classification. The algorithm differs markedly here from the Bayesian classifier and N-gram tagger in that it does *not* combine the scores for each member of the list found in the target context to be tagged, but rather uses only the single best piece of evidence available. See Step 7 for a discussion of this process.

Step 5: Optional Pruning and Interpolation

A potentially useful optional procedure is the interpolation of log-likelihood ratios between those computed from the full data set (the *global* probabilities) and those computed from the residual training data left at a given point in the decision list when all higher-ranked patterns failed to match (i.e. the *residual* probabilities). The residual probabilities are more relevant, but since the size of the residual training data shrinks at each level in the list, they are often much more poorly estimated (and in many cases there may be no relevant data left in the residual on which to compute the distribution of accent patterns for a given collocation). In contrast, the global probabilities are better estimated but less relevant. A reasonable compromise is to interpolate between the two, where the interpolated estimate is $\beta \times global + \gamma \times residual$. When the residual probabilities are based on a large training set and are well estimated, γ should dominate, while in cases the relevant residual is small or non-existent, β should dominate. If always $\beta = 0$ and $\gamma = 1$ (exclusive use of the residual), the result is a degenerate (strictly right-branching) decision tree with severe sparse data problems. Alternately, if one assumes that likelihood ratios for a given collocation are functionally equivalent at each line of a decision list, then one could exclusively use the global (always $\beta = 1$ and $\gamma = 0$). This is clearly the easiest and fastest approach, as probability distributions do not need to be recomputed as the list is constructed. Which approach is best? Using only the global proabilities does surprisingly well, and the results cited here are based on this readily replicatable procedure. The reason is grounded in the strong tendency of a word to exhibit only one sense or accent pattern per collocation (discussed in Step 7 and [Yarowsky, 1993]). Most classifications are based on a x vs. 0 distribution, and while the magnitude of the log-likelihood ratios may decrease in the residual, they rarely change sign. There are cases where this does happen and it appears that some interpolation helps, but for *this* problem the relatively small difference in performance does not seem to justify the greatly increased computational cost.

Two kinds of optional pruning can also increase the efficiency of the decision lists. The first handles the problem of "redundancy by subsumption," which is clearly visible in the example decision lists above (in WEEKDAY and *domingo*). When lemmas and word-classes precede their member words in the list, the latter will be ignored and can be pruned. If a bigram is unambiguous, probability distributions for dependent trigrams will not even be generated, since they will provide no additional information.

The second, pruning in a cross-validation phase, compensates for the minimal observed over-modeling of the data. Once a decision list is built it is applied to its own training set plus some held-out cross-validation data (*not* the test data). Lines in the list which contribute to more incorrect classifications than correct ones are removed. This also indirectly handles problems that may result from the omission of the interpolation step. If space is at a premium, lines which are never used in the cross-validation step may also be pruned. However, useful information is lost here, and words pruned in this way may have contributed to the classification of testing examples. A 3% drop in performance is observed, but an over 90% reduction in space is realized. The optimum pruning

strategy is subject to cost-benefit analysis. In the results reported below, all pruning except this final space-saving step was utilized.

Step 6: Train Decision Lists for General Classes of Ambiguity

For many similar types of ambiguities, such as the Spanish subjunctive/future distinction between *-ara* and *ará*, the decision lists for individual cases will be quite similar and use the same basic evidence for the classification (such as presence of nearby time adverbials). It is useful to build a general decision list for all *-ara/ará* ambiguities. This also tends to improve performance on words for which there is inadequate training data to build a full individual decision lists. The process for building this general class disambiguator is basically identical to that described in Steps 2-5 above, except that in Step 2, training contexts are pooled for all individual instances of the class (such as all *-ara/-ará* ambiguities). It is important to give each individual *-ara* word roughly equal representation in the training set, however, lest the list model the idiosyncrasies of the most frequent class members, rather than identify the shared common features representative of the full class.

In Spanish, decision lists are trained for the general ambiguity classes including *-o/-ó*, *-e/-é*, *-ara/-ará*, and *-aran/-arán*. For each ambiguous word belonginging to one of these classes, the accuracy of the word-specific decision list is compared with the class-based list. If the class's list performs adequately it is used. Words with idiosyncrasies that are not modeled well by the class's list retain their own word-specific decision list.

Step 7: Using the Decision Lists

Once these decision lists have been created, they may be used in real time to determine the accent pattern for ambiguous words in new contexts.

At run time, each word encountered in a text is looked up in a table. If the accent pattern is unambiguous, as determined in Step 1, the correct pattern is printed. Ambiguous words have a table of the possible accent patterns and a pointer to a decision list, either for that specific word or its ambiguity class (as determined in Step 6). This given list is searched for the highest ranking match in the word's context, and a classification number is returned, indicating the most likely of the word's accent patterns given the context[9].

From a statistical perspective, the evidence at the top of this list will most reliably disambiguate the target word. Given a word in a new context to be assigned an accent pattern, if we may only base the classification on a single line in the decision list, it should clearly be the highest ranking pattern that is present in the target context.

The question, however, is what to do with the less-reliable evidence that may also be present in the target context. The common tradition is to combine the available evidence in a weighted sum or product. This is done by Bayesian classifiers, neural nets, IR-based classifiers and N-gram part-of-speech taggers. The system reported here is unusual in that it does no such combination. *Only* the single most reliable piece of evidence matched in the target context is used. For example, in a context of *cote* containing *poisson*, *ports* and *atlantique*, if the adjacent feminine article *la cote* (the coast) is present, only this best evidence is used and the supporting semantic information ignored. Note that if the masculine article *le cote* (the side) were present in a similar maritime context, the most reliable evidence (gender agreement) would override the semantic clues which would otherwise dominate if all evidence was combined. If no gender agreement constraint were present in that context, the first matching semantic evidence would be used.

There are several motivations for this approach. The first is that combining all available evidence rarely produces a different classification than just using the single most reliable evidence, and when these differ it is as likely to hurt as to help. In a study comparing results for 20 words in a binary homograph disambiguation task, based strictly on words in local (±4 word) context, the following differences were observed between an algorithm taking the single best evidence, and an otherwise identical algorithm combining all available matching evidence:[10]

[9] If all entries in a decision list fail to match in a particular new context, a final entry called DEFAULT is used; it indicates the most likely accent pattern in cases where nothing matches.

[10] In cases of disagreement, using the single best evidence outperforms the combination of evidence 65% to 35%. This

Combining vs. Not Combining Probabilities

Agree -	Both classifications correct	92%
	Both classifications incorrect	6%
Disagree -	Single best evidence correct	1.3%
	Combined evidence correct	0.7%
Total -		100%

Of course that this behavior does not hold for all classification tasks, but *does* seem to be characteristic of lexically-based word classifications. This may be explained by the empirical observation that in most cases, and with high probability, words exhibit only one *sense* in a given collocation [Yarowsky, 1993]. Thus for this type of ambiguity resolution, there is no apparent detriment, and some apparent performance gain, from using only the single most reliable evidence in a classification. There are other advantages as well, including run-time efficiency and ease of parallelization. However, the greatest gain comes from the ability to incorporate multiple, non-independent information types in the decision procedure. As noted above, a given word in context may match several times in the decision list, once for its part of speech, lemma, inflected form, trigrams, and possibly word-class as well. By only using one of these matches, the gross exaggeration of probability from combining all of these non-independent log-likelihoods is avoided. While these dependencies may be modeled and corrected for in Bayesian formalisms, it is difficult and costly to do so. Using only one log-likelihood ratio without combination frees the algorithm to include a wide spectrum of highly non-independent information without additional algorithmic complexity or performance loss.

Evaluation:

Table 3 below gives a breakdown of performance on the comparative test set[11]. All of these evaluations were conducted with 5-fold cross-validation, using independent training and testing data.

TABLE 3: Decision List Performance

Pattern 1	Pattern 2	Agreement	BaseL	N
Spanish:				
anuncio	anunció	98.4%	57%	9459
registro	registró	98.4%	60%	2596
marco	marcó	98.2%	52%	2069
completo	completó	98.1%	54%	1701
retiro	retiró	97.5%	56%	3713
duro	duró	96.8%	52%	1466
paso	pasó	96.4%	50%	6383
regalo	regaló	90.7%	56%	280
terminara	terminará	82.9%	59%	218
llegara	llegará	78.4%	64%	860
esta	está	97.1%	61%	14140
mi	mí	93.7%	82%	1221
secretaria	secretaría	84.5%	52%	1065
French:				
cessé	cesse	97.7%	53%	1262
décidé	décide	96.5%	64%	3667
laisse	laissé	95.5%	50%	2624
commence	commencé	95.2%	54%	2105
côté	côte	98.1%	69%	3893
traité	traite	95.6%	71%	2865

observed difference is 1.9 standard deviations greater than expected by chance and is statistically significant.

[11]The French results are presented for reference only. Although all 3 algorithms have been applied to French data, space and logistical reasons have restricted the comparative evaluation to Spanish.

3 COMPARATIVE EVALUATION

A comparative analysis of system performance on the major ambiguity types found in Spanish is provided in the following table. The numbers are an average of the results for the test set of words presented in the preceding tables[12]. The common set of test cases helps highlight differences in performance between the 4 algorithms.

TABLE 4: Comparison of Performance on Spanish

Type	# 1 Baseline	# 2a N-gram (suffix)	# 2b N-gram (P.O.S.)	# 3 Bayesian Classifier	# 4 Decision List
-o/-ó	54.6	93.7	93.8	89.4	96.8
-ara/-ará	61.5	64.4	66.6	77.1	80.6
Function Word	81.8	89.3	89.9	84.3	95.4
Same POS	52.0	52.3	52.3	78.1	84.5

These results confirms several earlier hypotheses. First, the N-gram tagger and decision list are the best discriminators for *-o/-ó* and function word ambiguities, which involve primarily local, syntactic distinctions. Bayesian classifiers are less well suited for this task. In contrast, *-ara/-ará* ambiguities (of tense and mood), which involve longer range semantic dependencies, are best handled by decision lists followed by Bayesian classifiers. N-gram taggers perform very poorly on this task, as the distinguishing evidence is often beyond the immediate 3 word window. For ambiguities involving two words of the same part of speech, Bayesian classifiers and decision lists also perform best, while the part-of-speech-based N-gram tagger is not able to handle this case at all. Thus while the N-gram tagger and Bayesian classifier perform well on complementary subsets of the problem, the decision list algorithm performs well on both. It offers generality without apparent loss of precision.

Further analysis of these differences is presented below. However, before continuing with further comparison, it is important to note that all precision values in these experiments are based on *agreement* rates with the accent patterns in the test set, which themselves may be erroneous. Because we have only stripped accents artificially for testing purposes, and the "correct" patterns exist on-line in the original corpus, it is entirely objective and automatic to test performance, unlike in the evaluation of word-sense disambiguation and part-of-speech tagging, where at some point human judgements are required. Regrettably, however, there are errors in the original corpus, which can be quite substantial depending on the type of accent. For example, accents over the i (*í*) are frequently omitted, and in a sample test 3.7% of the appropriate accents were missing. Thus the previous results must be interpreted as agreement rates with the corpus accent pattern; the true percent correct may be several percentage points higher. The relatively low agreement rate on words with accented i's (*í*) in Tables 1, 2 and 3 are a result of these corpus errors. To study this discrepancy further, a human judge fluent in Spanish determined whether the corpus or decision list algorithm was correct in instances of two common ambiguities. For the ambiguity case of *mi/mí*, the corpus was incorrect in 46% of the disputed tokens. For the ambiguity *anuncio/anunció*, the corpus was incorrect in 56% of the disputed tokens. Although these appear to be extreme cases, they indicate limits to the precision of automatic evaluation.

At some point it would be interesting to pursue a more comprehensive evaluation of the accuracy of the corpus relative to the algorithm, and get a more precise estimate of algorithm "correctness" rather than just agreement with the corpus. However, this would require considerable effort, including multiple human judges and other mechanisms to reduce bias. Since automatic, objective scoring was one of the primary motivations for using the corpus accentings as the evaluation gold-standard, I would hope to find a more reliable corpus of test material instead. Nevertheless, initial

[12]In the case of the Bayesian Classifier, the performance from the optimal context width is used. Note that this context range was typically ± 2, the same as used by a trigram model.

results indicate that in some cases, the decision list system's precision may rival that of the AP Newswire's Spanish writers, translators, and copy-editors.

4 DISCUSSION AND CONCLUSION

The decision list algorithm presented here combines the strengths of both N-gram taggers and Bayesian classifiers, and outperforms both. Like the N-gram tagger, it utilizes trigram probabilities to model local syntactic constraints, and like the Bayesian classifier it successfully models long range lexical associations of a more semantic nature. By incorporating multiple types of evidence, the decision list not only exhibits generality and the ability to perform well on very different types of disambiguation problems, it also can exploit the additional available information to outperform the two competing algorithms on the tasks for which they are specialized.

It is often difficult to know in advance what information will be most useful for a particular discrimination task. Decision lists *consider* an extremely broad set of evidence in the training phase, but only utilize that which is most effective as a discriminating agent for the given task. While one could incorporate multiple sources of evidence in a Bayesian classifier as well, the key advantage of this decision list algorithm is that it allows the use of multiple, highly non-independent evidence types (such as root form, inflected form, part of speech, thesaurus category or application-specific clusters) and does so in a way that avoids the complex modelling of statistical dependencies. This allows the decision lists to find the level of representation that best matches the observed probability distributions. It is a kitchen-sink approach of the best kind – throw in many types of potentially relevant features and watch what floats to the top.

While there are certainly other ways to combine such evidence, the decision list approach has many advantages. The foremost is simplicity – it is extremely straightforward to use any new feature for which a probability distribution can be calculated. The algorithm, especially in its most basic form, is very easy to describe and implement. Other advantages are its perspicuity: the decision list is organized like a recipe, with the most useful evidence first and in highly readable form. It is much more comprehendable than an N-gram matrix or one of the impenetrable black boxes produced by many other machine learning algorithms. The generated decision procedure is easy to edit by hand, changing or adding patterns to the list. The algorithm is also readily applied to new domains: it was originally developed for homograph disambiguation in text-to-speech synthesis [Sproat et al., 1992], and was applied to the current problem without modification. The flexibility and generality of the algorithm and its potential feature set makes it readily applicable to other problems of recovering lost information from text corpora; I am currently pursuing its application to capitalization restoration and the task of recovering vowels in Hebrew text.

Overall, the decision list algorithm demonstrates considerable hybrid vigor, combining the strengths of N-gram taggers and Bayesian classifiers in a highly effective, general purpose decision procedure for lexical ambiguity resolution.

References

[1] Church, K.W., "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," in *Proceedings of the Second Conference on Applied Natural Language Processing, ACL*, 136-143, 1988.

[2] Gale, W., K. Church, and D. Yarowsky, "Discrimination Decisions for 100,000-Dimensional Spaces," Technical Memorandum, AT&T Bell Laboratories, 1992.

[3] Gale, W., K. Church, and D. Yarowsky, "A Method for Disambiguating Word Senses in a Large Corpus," *Computers and the Humanities*, 26, 415-439, 1992B.

[4] Kupiec, Julian, "Probabilistic Models of Short and Long Distance Word Dependencies in Running Text," in *Proceedings, DARPA Speech and Natural Language Workshop*, Philadelphia, February, pp. 290-295, 1989.

[5] Leacock, Claudia, Geoffrey Towell and Ellen Voorhees "Corpus-Based Statistical Sense Resolution," in *Proceedings, ARPA Human Language Technology Workshop*, 1993.

[6] Merialdo, B., 'Tagging Text with a Probabilistic Model," in *Proceedings of the IBM Natural Language ITL*, Paris, France, pp. 161-172, 1990.

[7] Mosteller, Frederick, and David Wallace, *Inference and Disputed Authorship: The Federalist*, Addison-Wesley, Reading, Massachusetts, 1964.

[8] Paul, D. B., "Speech Recognition Using Hidden Markov Models", in *The Lincoln Laboratory Journal*, 3, 1990.

[9] Rabiner, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", in *Proceedings of the IEEE*, 77, 257-285, 1989.

[10] Rivest, R. L., "Learning Decision Lists," in *Machine Learning*, 2, 229-246, 1987.

[11] Sproat, R., J. Hirschberg and D. Yarowsky "A Corpus-based Synthesizer," in *Proceedings, International Conference on Spoken Language Processing*, Banff, Alberta. October 1992.

[12] Tzoukermann, Evelyne and Mark Liberman, " A Finite-state Morphological Processor for Spanish," in *Proceedings, COLING-90*, Helsinki, 1990.

[13] Yarowsky, David "Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora," in *Proceedings, COLING-92*, Nantes, France, 1992.

[14] Yarowsky, David, "One Sense Per Collocation," in *Proceedings, ARPA Human Language Technology Workshop*, Princeton, 1993.

[15] Yarowsky, David, "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*, Las Cruces, NM, 1994.

Extracting a Disambiguated Thesaurus from Parallel Dictionary Definitions

Naohiko URAMOTO
IBM Research, Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken 242, Japan
uramoto@trl.vnet.ibm.com

Abstract

This paper describes a method for extracting disambiguated (bilingual) is-a relationships from parallel (English and Japanese) dictionary definitions by using word-level alignment. Definitions have a specific pattern, namely, a "genus term and differentia" structure; therefore, bilingual genus terms can be extracted by using bilingual pattern matching. For the alignment of words in the genus terms, a dynamic programming framework for sentence-level alignment proposed by Gale et al. [6] is used.

1 Introduction

Deeper and less ambiguous knowledge can be obtained by using parallel corpora than by using monolingual corpora. Research on this topic includes studies by Dagan et al. [4], who used parallel corpora for word selection in the target language in machine translation, and Utsuro et al. [17], who applied sample sentences in a English-to-Japanese dictionary to learning of case-patterns. Development of algorithms for sentence alignment in corpora is also a hot issue [2, 6, 10].

In this paper, bilingual sentences are taken from the IBM Dictionary of Computing [9] (originally written in English) and its Japanese translation [3]. Definitions in dictionaries have a restricted structure, namely, *genus term* and *differentia*. Using bilingual pattern matching, we obtain a bilingual pair consisting of an entry word and its genus term. It is assumed that the definitions in the dictionary and its translation have been aligned, since the matching between them is almost one-to-one, and the definitions are separated by entries, which makes it easy to align definitions. However, words in the genus terms for an entry must be aligned.

Most alignment algorithms require an *anchor point* that combines a part of one sentence and a part of sentence of other language. We use the bilingual pattern of the definitions. Use of the pattern makes it possible to align the words in part of a sentence without consulting a dictionary.

There is no doubt that a thesaurus is one of most useful sources of knowledge for semantic processing. The aim of our work is to develop a domain-dependent thesaurus for example-based disambiguation [16]. Much work has been done (for example, by Amsler [1], Klavans et al. [11], Nakamura et al. [14] and Guthrie et al. [7]) on the extraction of thesauruses from monolingual dictionaries such as the Longman Dictionary of Contemporary English (LDOCE) [15]. However, words in the definitions are ambiguous, and consequently it is difficult to get a disambiguated thesaurus. The advantage of using parallel texts is that it reduces the number of ambiguities inherent in each monolingual text, when the sentences in the texts are aligned. In our approach, the use of a bilingual dictionary makes it possible to acquire a set of

pairs of bilingual is-a relationships. The relationships are represented by [English word : Japanese Word] → [English hypernym : Japanese hypernym].

By simple matching using language-dependent patterns that appear in English and Japanese definitions, a genus term for an entry word can be extracted. The genus term consists of multiple words, and the alignment is not always one-to-one. However, in many cases, the order of words in English and Japanese genus terms is the same. Therefore, algorithms for sentence alignment can be applied to the problem. In this paper, the dynamic programming framework developed by Gale et al. [6] is used to align words in the genus term. In our alignment program, two preferences are used to measure the distance of an alignment. One is matching of syntactic categories, and the second is co-occurrence in other parts of the text.

2 Structure of the Definitions in a Parallel Dictionary

As a bilingual corpus, the IBM Dictionary of Computing [9] (written in English) and its Japanese translation [3] are used. Each contains about 10,000 entries for technical terms in the computer domain. Basically, one English definition is translated as one Japanese definition, and it is therefore easy to align sentences.

For example, the definitions of the entry word "active line" in the English version of the dictionary and its translation are as follows:

active line: 通信回線
(EDEF) a telecommunication line that is currently available for transmission of data.
(JDEF) 現在、データ転送に利用できる通信回線

The structure of definitions of on-line dictionaries has been analyzed (for example, in [14, 7, 18]). The main parts of the definition of a word are a *genus term* and a *differentia*. The genus term represents a hypernym of the entry word, and the differentia is used to distinguish the entry from other entries that have the same genus term.

In the English definition (EDEF), "telecommunication line" is the genus term for the entry "active line," and "that is currently available for transmission" is the differentia part. In the Japanese definition (JDEF), "通信回線" is the genus term, while "現在、データ転送に利用できる" is the differentia. The position of the genus term depends on the language. In English, it appears the beginning of the definition, while in Japanese it come at the end of the definition.

The following are the bilingual patterns for extracting the genus terms from the English and Japanese definitions:

(EPAT) PRE-DIFF GENUS-TERM+ POST-DIFF**
(JPAT) PRE-DIFF GENUS-TERM+*

The expression "WORD*" matches zero or more words, and "WORD+" matches one or more words. The expression GENUS-TERM matches words that have same syntactic category as an entry. PRE-DIFF matches a determiner. POST-DIFF matches a sequence that begins with a word whose category is not the same as that of the entry word. Information on the parts of speech of the words in the definitions is needed in order to recognize the genus words by using the patterns.

a_DET telecommunication_NOUN line_NOUN that_REL is_VERB available_ADJ for_PREP transmission_NOUN of_PREP data_NOUN

Figure 1: Tagged English definition for the entry "active line"

現在_ADV 、_PUNC データ転送_NOUN に_KJYO 利用_NOUN できる_JYODO 通信_NOUN 回線_NOUN

Figure 2: Tagged Japanese definition for the entry "active line"

2.1 Extraction of a Disambiguated Thesaurus

The genus term for an entry is extracted by using the bilingual pattern. The extraction procedure has three steps:

1. Part-of-speech tagging of parallel definitions
2. Matching of genus terms by using bilingual pattern
3. Alignment of the words in the genus terms extracted in step 2

First, the parts of speech of the definition sentences are tagged automatically. For English analysis, the English Slot Grammar (ESG) developed by McCord [13] is used. The Japanese Morphological Analyzer (JMA) developed by Maruyama et al. [12] is used for Japanese definitions. Figure 1 and 2 show the outputs of the English and Japanese taggers.

For each tagged parallel definition, the pattern described in Section 1 is applied. The result of matching is as follows:

Matching result of English definition:

ENG-ENTRY = active line
PRE-DIFF-1 = a_DET
GENUS-TERM-1 = telecommunication_NOUN
GENUS-TERM-2 = line_NOUN
POST-DIFF-1 = that_REL
POST-DIFF-2 = is_VERB
(other differentiae)

Matching result of Japanese definition:

JPN-ENTRY = 活動回線
PRE-DIFF-1 = 現在_ADV
PRE-DIFF-2 = データ_NOUN
(other differentiae)
GENUS-TERM-1 = 通信_NOUN
GENUS-TERM-2 = 回線_NOUN

In this case, there are two words for each entry. If the numbers of words in the genus terms are the same, the words match one-to-one, that is:

[active line:活動回線]
→ [[telecommunication:通信],[line:回線]]
→ [line:回線]

This is knowledge that represents bilingual and disambiguated is-a relationships between the entry word and its genus term. The relationships constitute a disambiguated thesaurus. If the numbers of words in the genus terms are different, the alignment procedure is required, which is described in Section 3.

2.2 Extracting a Disambiguated Thesaurus by Using Parallel Corpora

One of the issues in acquisition of is-a relationships from monolingual dictionaries is that words in the definitions contain ambiguities. Therefore, the words in the relationships must be disambiguated.

Use of parallel dictionaries makes it possible to extract disambiguated is-a relationships. For example, the parallel definition of the entry word "card column" is:

card column: カード欄
(E) a line of punch positions parallel to the shorter edge of a punch card.
(J) 穿孔カードの短い辺に平行な穿孔位置の行

From the definitions, the following relationships are extracted:

[card column:カード欄] → [line:行]

Both "card column" and "active line" have the genus term "line"; however, the meaning of "line" is different. The expression [line:行] represents the "line" in the definition means lines in images, while [line:回線] means electric lines. The granularity of word-sense is a serious problem affecting the acquisition of semantic knowledge. In this paper, a disambiguated word is presented by a translation pair such [line:行] or [line:回線]. The disambiguation level is useful when the knowledge is used for practical applications such as machine translation.

3 An Algorithm for the Extraction

3.1 Recognition of the Genus Terms of Entries

In Section 2, the acquisition of genus terms was described. Since the dictionary we used is for technical terms, the genus term often consists of multiple words. In this paper, the longest possible genus term for an entry is recognized. The genus term is a word sequence that contains the parts of speech of the entry, and also possibly adjectives, and adverbs. To absorb the differences between sets of parts of speech in English and Japanese, some modifications are needed:

- In English, the pattern "NOUN1 of NOUN2" in a genus term is transformed into "NOUN2 NOUN1." In Japanese, the pattern "NOUN1 の NOUN2" is transformed into "NOUN2 NOUN1".
- Adjectives and adverbs are treated as the same syntactic category.

The matching between words in genus terms is not always one-to-one. For example, suppose that the English genus terms "direct_ADJ addressing_VERB mode_NOUN" and the Japanese genus terms "直接_ADV アドレス_NOUN 指定_NOUN モード_NOUN" are aligned. The English pattern consists of three words, while the Japanese pattern consists of four words. For many-to-many matching, a method

for sentence alignment using dynamic programming framework developed by Gale et al. [5] is used. As they claim, the framework is useful when sequences such as sentences are compared by using a distance measure, which they calculate by using a probabilistic model. We use the framework to align the words in genus terms. As distance measure, we use the syntactic categories of the words and co-occurrence in the parallel dictionary and bilingual corpora.

4 A DP Algorithm for Genus Term Alignment

The algorithm for aligning words in the genus terms is basically the same as Gale's without the calculation of the distance measure, which we call the preference calculation.

Let ew(i) (i=0, ...) be the (i+1)th word in the English genus term, and let jw(j) (j=0, ...) be the (j+1)th word in the Japanese genus term. P(i,j) is the preference between the word sequences $ew(1), \ldots, ew(i)$ and $jw(1), \ldots, jw(j)$. Suppose that p is a preference function. For example, suppose that p(ew(1), jw(1); 1, 1) represents a match between ew(1) and jw(1) (one-to-one matching). P(i,j) is calculated according to the following formula:

P(i,j) = max(P1,P2,P3,P4,P5)
P1 = P(i-1,j-1) + p(ew(i-1),jw(j-1);1,1)
P2 = P(i-2,j-1) + p(ew(i-2),jw(j-1);2,1)
P3 = P(i-1,j-2) + p(ew(i-1),jw(j-2);1,2)
P4 = P(i-1,j) + p(ew(i-1),jw(j);1,0)
P5 = P(i,j-1) + p(ew(i),jw(i-1);0,1)

Suppose P(0,0) = 0. The preference function p reflects the following two factors:

- Alignment between words that have the same syntactic category is preferred.
- Alignment between words that co-occur in other sentences in the parallel dictionary or corpora is preferred.

The preference function p for alignment of k words from *jw(i)* and l words from *ew(j)* is calculated by using the following formula. The function *Syn_cat(w)* returns the syntactic category of the word w.

p(jw(i),ew(j);k,l) = category_preference(jw(i),ew(j),k,l) + co-occurrence_preference(jw(i),ew(j),k,l)

category_preference(jw(i),ew(j),k,l) =

$$\begin{cases} 1 & : (k = 1 \textit{ and } l = 1) \textit{ and } (syn\_cat(jw(i)) = syn\_cat(ew(j))) \\ 0.75 & : (k = 1 \textit{ and } l = 1) \textit{ and } (syn\_cat(jw(i)) \; != \; syn\_cat(ew(j))) \\ 0.5 & : k = 2 \textit{ or } l = 2 \\ 0 & : k = 0 \textit{ or } l = 0 \end{cases}$$

co-occurrence_preference(jw(i),ew(j),k,l) =

$$\begin{cases} \frac{n}{m} (m > 0) : \textit{Here, } n \textit{ is the number of definitions that contain the same alignment of words but do not contain the same differentiae, while } m \textit{ is the number of the total number of definitions that contain the same alignment of words.} \\ 0 : (m = 0) \end{cases}$$

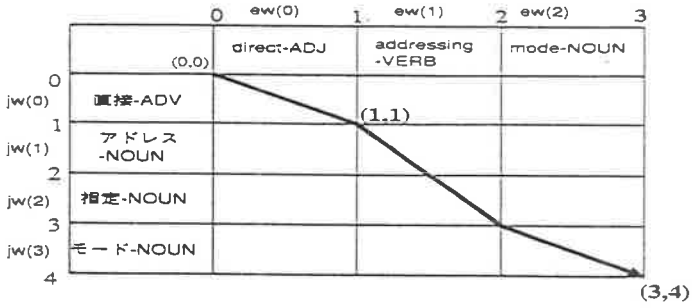


Figure 3: Alignment Matrix

For example, p(jw(1),ew(1);2,1) gives the preference for matching of one English word, "addressing," and two Japanese words, "アドレス" and "指定". The matching is one-to-two, so the category_preference is 0.5. For the co-occurrence_preference, the following bilingual definition is found among the definitions:

(E) `ACF/TCAM, any point-to-point line configuration in which the station on the line does not use polling and <<addressing>> characters.`
(J) ACM/TCAM において、回線上のステーションがポーリングと<<アドレス指定>>文字を使用しないポイントツーポイント回線構成

If the number of the definition that contains the words "addressing" is two, the preference is $\frac{1}{2}$. Therefore p(jw(1),ew(1);2,1) is $0.5 + 0.5 = 1.0$.

To align the words, an alignment matrix is created. Figure 3 shows the matrix for the example. Rows in the matrix show the sequence of English words, and columns represent the sequence of Japanese words. The position (i,j) in the matrix represents P(i,j). The path from (0,0) to (3,4) in the matrix represents the alignment of the words.

In this case, the shortest path is [(0,0)→(1,1)→(2,3)→(3,4)], which gives a correct alignment.

5 Experiments

5.1 Experiment-1: Extraction from a Parallel Dictionary

We concluded a small experiment on extraction of a disambiguated thesaurus of 1,000 pairs of definitions for entries that begin with "a". The matching of genus terms in definitions was done by a grep-like tool is called parallel grep (PGREP). As options, PGREP requires English and/or Japanese patterns and actions during pattern matching. If the matching of the genus term was not one-to-one, the words were aligned by dynamic programming. The results of the alignment were compared with a human's answers. We obtained a correct alignment rate of 91.3 % for the 1,000 sample definitions.

The main cause of failure was the difference in word order of English and Japanese genus terms. Another problem is that the pattern "WORD1 WORD2 of WORD3," which is very common in the definitions, is translated in various ways.

5.2 Experiment-2: Extraction from a Parallel Corpora

In experiment-1, we extracted is-a relationships from definitions that have a specific pattern. However, since there are few parallel dictionaries, we had to extract the relationships from "ordinary" bilingual

corpora. In our second experiment, we used bilingual (English and Japanese) computer manuals. Hearst proposes a method for extracting is-a relationships from a monolingual encyclopedia by using patterns "such NP as NP" and "NP, including NP" [8]. We use the simpler pattern "NOUN is a/the GENUS-TERM." The pattern cannot be used for monolingual text, since the many meanings of "be" cause ambiguities. However, by using both English and Japanese patterns, the number of ambiguities can be reduced, since some ambiguities in the sentences are resolved in their translation [17].

In the experiment, the following patterns are used (the expression "|" is an OR operator).

English pattern:
MOD* E-NOUN+ [is a|is the|are] MOD* E-GENUS-TERM*

Japanese pattern:
MOD* J-NOUN+ [は | が | とは] MOD* J-GENUS-TERM+ [です。 | で、 | である。 |。 | を言います。]

By using the pattern bilingual pair [E-NOUN:J-NOUN] → [E-GENUS-TERM:J-GENUS-TERM] was extracted.

For example, from the following sentences in the manual, the relationship [offset:オフセット] → [number:数] is extracted.

(E) If the <offset$_{E-N}$> *is a* negative <<number $_{E-GT}$>>, the routine associated with the offset probably did not allocate a save area, or the routine may have been called using SVC-assisted linkage.
(J) <オフセット$_{J-N}$>が負の<<数$_{J-GT}$>>である場合には、このオフセットに関連づけられるルーチンが保管域を割り振らなかったか、そのルーチンがＳＶＣ援助連係を使用して呼び出された可能性があります。

These patterns were applied to a bilingual manual text containing 2,000 pairs of sentences. Thirty-four pairs were matched of which 30 were correct. Most failures were caused by free translation.

6 Related Work

For the practical use of the alignment program, the following four issues concerning its accuracy must be considered.

1. Robustness when used with very large corpora
2. Use of practical computational resources
3. Language-independence of the algorithm
4. Use of information solely in the corpora to be aligned.

Since our approach uses simple pattern matching and dynamic programming for some words in sentences, the first two obstacles can be avoided. In our framework, information on the syntactic categories and co-occurrence of the words in the corpora is used. Though the set of syntactic categories needed depends on the language, our method can be applied if some heuristics are used.

In terms of knowledge acquisition from dictionaries, use of a parallel dictionary makes it possible to construct disambiguated (bilingual) relationships. One of the obstacles in the monolingual approach is ambiguity of words in definitions. Guthrie et al. proposed a method for extracting disambiguated is-a relationships from the LDOCE with matching of case-patterns [7]. However, since the number of semantic markers used in the LDOCE is relatively small, it is difficult to resolve ambiguities completely.

Our method uses bilingual definitions, so the word-senses of the words in the thesaurus are represented by pairs of English word and its Japanese translation. The notation is still ambiguous when the words in both languages contain the same ambiguities. However, the notation is useful when the knowledge is used in practical applications such as machine translation.

There have been some studies of the alignment of sentences [2, 6, 10]. These studies are classified according to what kind of "anchor point" binds both (parts of) sentences. Kay et al. use words appearing in the sentences [10]. Gale et al. use the lengths of sentence pairs [6], while Brown et al. employ the numbers of words in them [2]. In word-to-word alignment, the fact that the word order depends on the language prevents the development of a word-to-word alignment algorithm. In this paper, alignment of genus words that form a part of sentence is proposed. The framework developed for sentence alignment can be applied to our work, using the "genus term and differentia" structure in the definition sentences.

7 Conclusion

We have described a method for extracting disambiguated (bilingual) is-a relationships from parallel (English and Japanese) dictionary definitions by using word-level alignment: We are now evaluating the method by using more examples. Alignment of English and Japanese is more difficult than that of English and, say, French, since Japanese phrases have no word boundaries. Therefore, the recognition of compound nouns depends on the lexicon and the algorithm in the tagger. For example, it is clear that "telecommunication line" consists of two words. However, its Japanese translation "通信回線" can be recognized as one word or two words ("通信" and "回線"). To absorb the differences between the languages, a more refined algorithm is required.

In the work described here, only genus terms were extracted. However, definitions contain other useful information. The alignment of other parts of sentences is important. Knowledge extraction from an unrestricted bilingual corpus rather than from definition sentences is another challenging issue. Tools for dealing with bilingual data are also needed.

References

[1] R. A. Amsler. "A Taxonomy for English Nouns and Verbs". In *Proceedings of the 19th Annual Meeting of the ACL*, pages 133–138, 1981.

[2] P. F. Brown, J. C. Lai, and R. L. Mercer. "Aligning Sentences in Parallel Corpora". In *Proceedings of the 29th Annual Meeting of ACL*, pages 169–176, 1991.

[3] IBM Corporation. *IBM Dictionary of Computing (in Japanese)*, volume N:SC20-1699-07. IBM Corporation, 1991.

[4] I. Dagan, A. Itai, and U. Schwall. "Two Languages are More Informative Than One". In *Proceedings of ACL-91*, 1991.

[5] W. Gale and K. W. Church. "A Program for Aligning Sentences in Bilingual Corpora". In *Proceedings of ACL-91*, pages 177–184, 1991.

[6] W. A. Gale and K. W. Church. "A Program for Aligning Sentences in Bilingual Corpora". *Computational Linguistics*, 19(1):75–102, 1993.

[7] L. Guthrie, B. M. Slator, Y. Wilks, and R. Bruce. "Is There Content in Empty Heads?". In *Proceedings of COLING-90*, pages 138–143, 1990.

[8] M. A. Hearst. "Automatic Acquisition of Hyponyms from Large Text Corpora". In *Proceedings of COLING-92*, pages 539–545, 1992.

[9] IBM Corporation. *IBM Dictionary of Computing*, volume SC20-1699-07. IBM Corporation, 1988.

[10] M. Kay and M Roscheisen. "Text-Translation Alignment". *Computational Linguistics*, 19(1):121–142, 1993.

[11] J. Klavans, M. S. Chodorow, and N. Wacholder. "From Dictionary to Knowledge Base via Taxonomy ". In *Proceedings of the 6th Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary and Text Research*, pages 110–127, 1990.

[12] H. Maruyama and S. Ogino. "The Mega-Word Tagged Corpus Project". In *Proceedings of TMI-93*, 1993.

[13] M. McCord. "The Slot Grammar System". Technical Report RC17313, IBM Research Report, 1991.

[14] J. Nakamura and M. Nagao. "Extraction of Semantic Information from an Ordinary English Dictionary and Its Evaluation". In *Proceedings of COLING-88*, pages 459–464, 1988.

[15] P. Procter. *"Longman Dictionary of Contemporary English"*. Longman Group Limited, Harlow and London, England, 1978.

[16] N. Uramoto. "Lexical and Structural Disambiguation Using an Example-Base". In *The 2nd Japan-Australia Joint Symposium on Natural Language Processing*, pages 150–160, 1991.

[17] T. Utsuro, Y. Matsumoto, and M. Nagao. "Lexical Knowledge Acquisition from Bilingual Corpora". In *Proceedings of COLING-92*, pages 581–588, 1992.

[18] Y. Wilks, D. Fass, C. Guo, J. McDonald, T. Plate, and B. Slator. "Providing Machine Tractable Dictionary Tools". *Machine Translation*, 5:99–154, 1990.

Application of Corpora in Second Language Learning
— The Problem of Collocational Knowledge Acquisition —

Kenji KITA †, *Takashi OMOTO* †, *Yoneo YANO* † *and Yasuhiko KATO* ‡

† Department of Information Science and Intelligent Systems
Faculty of Engineering
Tokushima University
Tokushima 770, JAPAN
e-mail: kita@is.tokushima-u.ac.jp

‡ Section for Dictionary Research
The National Language Research Institute
Kita-ku, Tokyo 115, JAPAN
e-mail: kateaux@tansei.cc.u-tokyo.ac.jp

Abstract

While corpus-based studies are now becoming a new methodology in natural language processing, second language learning offers one interesting potential application. In this paper, we are primarily concerned with the acquisition of collocational knowledge from corpora for use in language learning. First we discuss the importance of collocational knowledge in second language learning, and then take up two measures, mutual information and cost criteria, for automatically identifying or extracting collocations from corpora. Comparative experiments are made between the two measures using both Japanese and English corpora. In our experiments, the cost criteria measure proved more effective in extracting interesting collocations such as fundamental idiomatic expressions and phrases.

1 Introduction

Recent rapid advances in computer technology (particularly the advent of large storage devices and parallel computers) and numerous data collection efforts have caused a shift in natural language applications from a knowledge-based to a corpus-based or data-intensive approach. The knowledge-based approach focused on abstraction of language, describing linguistic phenomena through minimal core knowledge such as parts-of-speech, syntactic and semantic rules. Linguistic phenomena, however, vary so vastly that they cannot be described through core knowledge. In addition, hand-coding knowledge takes a lot of time and hard work. The knowledge-based approach, therefore, has been found wanting in developing large-scale practical NLP systems.

On the other hand, the corpus-based approach makes no claim about the compactness of the knowledge. Rather, the corpus-based approach derives more power from massive quantities of textual data than from hand-coded knowledge, being able to compensate for the weakness of the knowledge-based approach through authentic examples and various statistics of language use. With the availability of large corpora in recent years, many successful results have been derived from corpus-based studies. These include part-of-speech tagging [Kupiec 1992], parsing [Magerman and Marcus 1990], example-based machine translation[Sumita and Iida 1992], statistical machine translation [Brown et al. 1990, Brown et al. 1993], language modeling [Jelinek 1990, Kita 1992] and many other related areas.

One interesting potential use of corpora is for second language learning. Kita et al. [Kita et al. 1993b] discussed various way of using corpora in language learning. The greatest advantage of using corpora in language learning is that the corpora provide a body of evidence for the function and usage of words and expressions. At the same time, deriving lexical knowledge from large-scale corpora via automated procedures, as well as its use in language learning CAI systems, is one of the most important issues.

In this paper, we are primarily concerned with the acquisition of collocational knowledge from corpora. The organization of this paper is as follows. Section 2 gives an overview of corpus-based CALL (Computer-Assisted Language Learning). In Section 3, we describe why collocational knowledge is important in second language learning. In Section 4, we discuss the automatic extraction of collocations, taking up two measures, mutual information and cost criteria, for identifying or extracting collocations from corpora. In Section 5, we describe comparative experiments in extracting collocations and discuss the two measures.

2 The Use of Corpora in Second Language Learning

There have been many language learning systems developed so far. Of course, the goal of creating language learning systems is to have learners master practical language skills. In spite of efforts by many researchers, we are still quite far from this goal although we admit that partial success has been achieved. Why? First, language learning systems developed so far are too domain-limited, i.e. they operate only on a restricted subject matter or purpose and accept sentences only of a limited or restricted nature. Second, researchers paid attention to knowledge representation models themselves rather than to the knowledge to be entered. In consequence, systems lack wide coverage and robustness, being often called "toy systems".

Corpus-based CALL offers great possibilities in building practical language learning systems. Some topics of corpus-based CALL includes:

- **Linguistic knowledge acquisition from corpora.**

  Language learning systems must incorporate many kinds of linguistic knowledge. Usually, the linguistic knowledge is hand-coded by humans. The resulting knowledge, however, sometimes does not match real-world usage. Also, hand-coding knowledge takes a lot of time and hard work. The current availability of large computer-readable corpora presents the possibility of deriving knowledge via automated procedures. Incorporating the derived knowledge makes language learning systems quite useful for dealing with unrestricted texts, making systems eminently robust.

- **Enhancement of translation skills through bilingual corpora.**

  Bilingual corpora consist of parallel texts which content is essentially equivalent. Thus, they have potential possibilities in enhancing learners' translation skills.

- **Enhancement of oral/aural skills through speech corpora.**

  There are corpora in which actual speech data are encoded. Speech corpora can be used for enhancing listening/speaking abilities. In particular, for full development of aural comprehension ability, speaker-dependent practice (using speech from one speaker) does not suffice; it is necessary to provide learners with extensive listening practice using speeches from many speakers. For that purpose, speech corpora is indispensable for language learning systems.

- **Multimedia language learning through structured corpora.**

  Learning environment with multimedia aids is one of the recent increasing interests. As stated above, corpus-based language learning enables us to use not only texts but also speech material. Moreover, a corpus encoded with SGML can be used to link words of a text to images of its objects. Thus, multimedia language learning which integrates texts, speech and images would be possible.

- **Retrieving examples from corpora.**

  Learners often want to know how a word is used within a sentence. Although dictionaries are fine for that purpose, they include only typical examples. A corpus includes quantitatively a sufficiently large amount of examples of a language, providing a more extensive usage of words. In addition, various computational tools have been developed for retrieving examples from corpora.

- **Augmenting incomplete knowledge with many examples.**

  Language learning systems are required to accept sentences from learners, where input sentences are judged correct or not through the process of parsing. To do this, a traditional approach uses phrase-structure grammars, sometimes augmented by sematic information. However, a grammatical approach often does not work out because grammars inherently contain problems such as the *overgeneration problem*, the *undergeneration problem*, and the *ambiguity problem*. A corpus includes many examples, being able to compensate the incompleteness of a grammar through actual examples and statistical data.

3 Importance of Collocational Knowledge in Language Learning

There has been much theoretical and applied research on collocations, both from a linguistic and an engineering point of view. Consequently, the definition of collocation differs according to the researcher's interest and standpoint. This paper adopts the most comprehensive definition: a collocation is a cohesive word cluster, including idioms, frozen expressions and compound words.

The importance of collocations has been stressed in an extensive literature. From a language learning viewpoint, it can be summarized as follows:

- In language learning, learners must pay attention to how words are used rather than to individual words by themselves. Collocational knowledge indicates which words co-occur frequently with other words and how they combine within a sentence. Therefore, collocational knowledge is especially effective in sentence generation [Smadja and McKeown 1990, Smadja 1993].

- Collocational knowledge is very difficult to acquire for second language learners. A typical example is the pair of words "strong" and "powerful" [Church et al. 1991, Smadja 1991]. These two words have similar meanings, but their usage is quite different. For example, native English speakers prefer saying "strong tea" to "powerful tea", and prefer saying "powerful car" to "strong car". For non-natives, however, it is difficult to catch the subtle distinctions between these two words. These lexical preferences were sometimes ignored in the traditional knowledge-based approach; nevertheless they are the most important source for word choice and word ordering.

- It is pointed out that human translation process is based on analogical thinking [Nagao 1984]. First, a human translator properly decomposes a sentence into certain fragmental phrases,

then s/he translates each fragmental phrase by analogy with other examples, and finally composes fragmental translations into one sentence. Collocations are suitable for fragmental translation units.

- From a cognitive point of view, it is said that human language acquisition is governed by the law of maximal efficiency [Wolff 1991]. In other words, data compression, often called chunking, is performed to minimize storage demands in the brain. A chunk is considered to be a pattern which repeatedly appears in a variety of contexts. Collocations are good candidates for chunk units.

4 Extracting Collocations from Corpora

In the past, several approaches have been proposed to extract collocations from corpora. Church et al. [Church and Hanks 1990, Church et al. 1991] introduced the association ratio, which indicates how strongly two words are related, based on the information-theoretic concept of mutual information. Smadja et al. [Smadja and McKeown 1990, Smadja 1991, Smadja 1993] take into account word distance as well as word strength for a measure of word association. Also, Basili et al. [Basili et al. 1992] proposed a syntax-based approach. Particularly, mutual information plays a central role in recent lexical statistical research. To take a few examples, Hindle and Rooth [Hindle and Rooth 1993] applied mutual information to disambiguate prepositional phrase attachments, and Brown et al. [Brown et al. 1992] used it in determining word classes.

In this section, after surveying how mutual information can be used to extract collocational information, we introduce another measure, called *cost criteria* [Kita et al. 1993], to automatically extract interesting collocations from corpora. Comparative experiments and discussions will be described in the next section.

4.1 Mutual Information

The mutual information between two words x and y is defined as follows [Church and Hanks 1990, Church et al. 1991]:

$$I(x,y) = \log \frac{P(x,y)}{P(x)P(y)} \tag{1}$$

Here, $P(x)$ and $P(y)$ are word occurrence probabilities, and can be estimated from the number of occurrences of the words, $f(x)$ and $f(y)$, and the number of words in the corpus, N.

$$P(x) = \frac{f(x)}{N} \quad \text{and} \quad P(y) = \frac{f(y)}{N} \tag{2}$$

$P(x,y)$, the joint probability of x and y, is estimated in a similar way.

$$P(x,y) = \frac{f(x,y)}{N} \tag{3}$$

where $f(x,y)$ is the number of occurrences of x followed by y.

The mutual information $I(x,y)$ compares the probability of observing x and y together with the probabilities of observing x and y simply by chance. Thus, a large value indicates that the two words x and y have a strong relationship. By extracting word pairs with large mutual information values, we can obtain common collocations.

Because mutual information values are defined for two words, this simple method can only extract collocations of length two. However, a generalization is suggested in [Jelinek 1990] as follows:

1. Start out from the basic vocabulary V_0. Set $n = 0$.
2. Augment the vocabulary V_n by all word sequences "$x\ y$" for which $I(x,y) > Thr$, where Thr is a predetermined threshold.
3. From Step 2, a new vocabulary V_{n+1} is established.
4. Adjust the counts to reflect the new vocabulary V_{n+1}.
5. Resume from Step 1 with V_{n+1} as its basis.

With this iterative procedure, the final vocabulary includes collocations of arbitrary length.

4.2 Cost Criteria

The cost criteria measure is based on the assumptions that (1) collocations are recurrent word sequences, and (2) the recurrent property is captured by the absolute frequency of a word sequence. However, a simple absolute frequency approach does not work, because the frequency of a sub-sequence is always higher than that of the original word sequence. For example, because "in spite" is a sub-sequence of "in spite of", "in spite" appears more frequently than "in spite of". However, given the context "in spite", it is highly probable that "of" follows "in spite". Consequently, we must consider that "in spite of" is a collocation but "in spite" is not. The idea of cost criteria formalizes this, and it can quantitatively estimate the extent to which processing is reduced by considering a word sequence as one unit.

Before the presentation of a formal definition, we introduce the following notation:

$$\alpha \quad \cdots \quad \text{a word sequence.} \tag{4}$$

$$|\alpha| \quad \cdots \quad \text{the length of } \alpha. \tag{5}$$

(the number of words in α)

$$f(\alpha) \quad \cdots \quad \text{number of occurrences of } \alpha \text{ in a corpus.} \tag{6}$$

We define $K(\alpha)$, the cost reduction incurred by handling α as a unit:

$$K(\alpha) = (|\alpha| - 1) \times f(\alpha) \tag{7}$$

$K(\alpha)$ is interpreted as follows. Assume here that, in the corpus, there exists a word sequence α, which is composed of $|\alpha|$ words and occurs $f(\alpha)$ times. Also assume that the cost of processing one word is 1. Similarly, when processing α as a single unit, its processing cost is 1. If a word sequence is processed one word at a time, it is reasonable to assume that the processing cost is proportional to the length of the word sequence. That is, the processing cost for α is $|\alpha|$. By considering α as one unit, the processing cost is reduced by $|\alpha| - 1$. Since α appears $f(\alpha)$ times, we can conclude that the total cost reduction becomes $(|\alpha| - 1) \times f(\alpha)$, which is the definition of $K(\alpha)$.

In reality, however, the problem is not so simple, because word sequences are not mutually disjoint. Consider the case where a word sequence α is a sub-sequence of β (for example, $\alpha =$ "in spite", $\beta =$ "in spite of"). Then, we have:

$$f(\alpha) \geq f(\beta) \tag{8}$$

Further, the word sequence α, $f(\beta)$ times out of $f(\alpha)$ times, will be identified as β. Thus, the actual cost reduction for α is defined as:

$$K(\alpha) = (|\alpha| - 1) \times (f(\alpha) - f(\beta)) \tag{9}$$

Finally, we can extract collocations from a corpus by the following steps:

1. Calculate $K(\alpha)$ for each word sequence α in a corpus.
2. Rank a word sequence α by using the value $K(\alpha)$.
3. Extract higher rank word sequences as collocation candidates.
4. Re-calculate $K(\alpha)$ for each α in the collocation candidates.

5 Experiments and Discussions

5.1 The ADD Corpus

In our experiments, the ADD (ATR Dialogue Database) Corpus [Ehara et al. 1990] created by ATR Interpreting Telephony Research Laboratories in Japan was used. The ADD Corpus is a large structured database of dialogues collected from simulated telephone or keyboard conversations which are spontaneously spoken or typed in Japanese or English. This corpus consists of parallel texts of Japanese and English, aligned by utterance. Also, sentences in ADD are morphologically analyzed and annotated with various kinds of syntactic, semantic, and phonological information.

Currently, the ADD Corpus contains textual data from two tasks (text categories); one consists of simulated dialogues between a secretary and participants at international conferences (Conference Task), and the other of simulated dialogues between travel agents and customers (Travel Task).

In our experiments, we used the keyboard dialogues from the Travel Task, which include approximately 120,000 Japanese words and 100,000 English words. The telephone dialogue include linguistic phenomena, such as filled pauses ("ah", "uh", etc.), restarts (repeating a word or phrase) and interjections, so we did not use them for our experiments.

5.2 Results and Discussions

Figure 1 shows some interesting Japanese collocations extracted using respectively mutual information and cost criteria. Figure 2 shows some English ones.

Before discussing the results, we first overview the characteristics of Japanese phrases. In general, the order of major constituents in a Japanese sentence is rather free. However, predicate phrase positioning is dominated by the so-called predicate-phrase ending constraint: a predicate phrase appears at the end of its clause. Furthermore, a predicate phrase often has a complex form, consisting of a main predicate such as a verbal noun, verb or adverb, combinations of auxiliary predicates, and a sentence-final particle. These auxiliary predicates and sentence-final particles add various complementary meanings to a sentence, such as honorific, causative, and prohibitive meanings, etc.

As can be seen from the experimental results (Figure 1), the method based on mutual information tends to extract compound noun phrases, while cost criteria tends to extract complex predicate phrase patterns. Almost all the collocations extracted are in this category. For example,

Mutual Information	Cost Criteria
ichi ryuu no orchestra ni yoru ensou	desho u ka
Jouzankei-onsen to set ni nat ta golf-pack	desu ka
Kunitachi-shi Ishida	desho u
chijou e	mashi ta
night-tour ya dinner-show	sou desu
kaihatsu ga sakan	sou desu ka
buchou ya kachou	to iu koto
6 mai tsuzuri	sou desu ne
hizuke henkou sen wo koe	masu ka
moushikomi kin toshite o azukari	desu ne
Shinjuku-ku Naitou-chou 1 banchi	o negai shi masu
kokunai sen no daiya	itashi masu
hakkou kaisha ni teishutsu	o negai itashi masu
Kenya Tanzania Safari	to omoi masu
yuuran sen no senchou	te ori masu
kaisui yoku	tai no desu ga
yuukyuu kyuuka	wakari mashi ta
Matsushima-wan meguri	kashikomari mashi ta
resort kaihatsu	ni nari masu
umi to yama	to iu no ha
yuujin no hahaoya	shi tai no desu ga
Hachiman-daira Towada Hiraizumi	to iu koto de
danjo betsu no uchiwake	na n desu ga
hakubutsu kan	shi tai no desu
dou nenpai	shouchi itashi mashi ta
senmon yougo	to iu koto desu
yuukou kigen	sou na n desu
genkin kakitome	arigatou gozaimashi ta
Shanghai Sian	sa se te itadaki masu
Setagaya-ku Kyoudou	o mata se itashi mashi ta
seinen gappi	sou na n desu ka
moyori no eki	shitsurei itashi masu
choushoku to chuushoku	yoroshii desho u ka
Fuji-ginkou honten	ka mo shire mase n
gouka kyakusen	irasshai masu ka

Figure 1: Some examples of extracted collocations. (Japanese)

Mutual Information	Cost Criteria
yacht harbor	is that so
Echigo Yuzawa	thank you very much
Fifth Avenue	I would like to
General Affairs	I see
Mitsuboshi trading	my name is
slide projector	sorry to have kept you waiting
strong background	is that right
cross the International Date	in that case
the F1 Grand Prix	I understand
Shiretoko Sightseeing Boat Inc.	thank you for
it's my pleasure	do you have any
at the Hotel New Tanda	good bye
give a speech	would you like to
head Mr. Kuwata	I am very sorry
wine production	a little
Wall Street	be able to
jazz dance	I got it
my mother in law	I'll be waiting for your call
to the historic sites	may I have your name and address
I am not that familiar	how much
Keirin and Peking	all right
cause the inconvenience	as soon as possible
holding a paper	then would you give me your
baths and toilets	the other day
Las Vegas	make the reservations
Queen Elizabeth	a lot of
Main Branch	I will call you
Sales Department	that's right
self introduction	how about
zip code	at that time
international cards	the application fee
to the Grand Canyon	is that okay
The Hyatt Regency	I appreciate your
flight number JS	of course
Canadian Rockies and Vancouver	so please hold the line

Figure 2: Some examples of extracted collocations. (English)

the collocations "desho u ka" and "desu ka", which had a high cost reduction, are used very often to make interrogative sentences in Japanese. The collocation "tai no desu ga" is usually used to express a speaker's request, whose meaning is "(I) would like to".

Considering that beginners in the Japanese language are sometimes annoyed by the complex conjugation properties of predicate constituents, it is educationally effective to provide them with typical and frequently used predicate phrase patterns. In that respect, we can say that the cost criteria measure is superior to mutual information.

The comments above are also true of the English data. Mutual information tends to extract compound noun phrases, while cost criteria tends to extract frozen phrase patterns such as "thank you very much" and "I would like to".

Why does mutual information fail to extract these patterns? Here, let us take "I will" as an illustrative example, which has been picked out by cost criteria ("I will" is omitted from Figure 2) but not by mutual information. In our corpus, "I" occurs 2,907 times, "will" occurs 920 times, and "I will" occurs 264 times. Therefore, we have

$$\begin{aligned} I(\mathrm{I}, \mathrm{will}) &= \log \frac{\frac{264}{100{,}000}}{\frac{2907}{100{,}000}\,\frac{920}{100{,}000}} \\ &= 3.3 \end{aligned} \tag{10}$$

This value is not so large, so the two words "I" and "will" cannot be considered to have a significant relationship.

According to the same reasoning, patterns such as "I would like to" and "thank you very much" are excluded as collocation candidates. However, in the ADD Corpus, more than fifty per cent of the sentences that involve the word "would" are subsumed under the pattern "(I) would like to ∼". Therefore, this pattern should be included in the collocation list.

Another drawback using mutual information is the sparseness of data. A corpus cannot provide sufficient data about every word-word relationship. Some word pairs may have high mutual information values in spite of their low frequency in the corpus. For example, the first ranked collocation was "yacht harbor", which occurs only twice in the ADD Corpus. On the contrary, since the cost criteria measure is based on absolute frequency, such phenomena never happens.

Furthermore, because the cost criteria measure estimates the extent to which processing is reduced, it can be considered to be a model of learners' work load. Also, collocations extracted using cost criteria can cover a wide range of human linguistic behavior. To sum up, we can say that the cost criteria measure is more suitable from the viewpoint of language learning.

6 Conclusion

With the growing availability of large textual resources, corpus-based studies are gaining more and more attention among computational linguists and computer scientists. In Particular, automatic acquisition of lexical knowledge from corpora is one of the most important and interesting issues. In this paper, we have taken up the problem of how to acquire collocational knowledge and discussed its importance for language learning. We have also presented an effective measure, called cost criteria, for automatic extraction of collocations from corpora. Comparative experiments with mutual information have shown that the cost criteria measure is more suitable for the purpose of language learning.

Unfortunately, the current implementation can only extract collocations of uninterrupted word sequences. Our next plan is to refine the method to extract collocations of interrupted sequences, and to utilize lexical information such as parts-of-speech in order to prevent an improper word sequence from being recognized as a collocation. Also, we hope to incorporate extracted collocations into a language learning CAI system.

Acknowledgments

The idea of cost criteria was developed while the first author was staying at ATR Interpreting Telephony Research Laboratories. The authors are deeply grateful to co-researchers in ATR, Kentaro Ogura (currently with NTT Network Information Systems Laboratories) and Tsuyoshi Morimoto, for their fruitful discussions and comments. The authors are also grateful to the members of our laboratory in Tokushima University for their help and encouragement. Special thanks to Gerardo Ayala, Ingrid Kirschning and John Phillips for reading the manuscript.

References

[Basili et al. 1992] Basili, R., Pazienza, M. T. and Velardi, P.: "A shallow syntactic analyzer to extract word associations from corpora", *Literary and Linguistic Computing*, Vol. 7, No. 2, pp. 113-123, 1992.

[Brown et al. 1990] Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L. and Roossin, P. S.: "A statistical approach to machine translation", *Computational Linguistics*, Vol. 16, No. 2, pp. 79-85, 1990.

[Brown et al. 1992] Brown, P. F., Pietra, V. J. D., deSouza, P. V., Lai, J. C. and Mercer, R. L.: "Class-based n-gram models of natural language", *Computational Linguistics*, Vol. 18, No. 4, pp. 467-479, 1992.

[Brown et al. 1993] Brown, P. F., Pietra, S. A. D., Pietra, V. J. D. and Mercer, R. L.: "The mathematics of statistical machine translation: Parameter estimation", *Computational Linguistics*, Vol. 19, No. 2, pp. 263-311, 1993.

[Church and Hanks 1990] Church, K. W. and Hanks, P.: "Word association norms, mutual information, and lexicography", *Computational Linguistics*, Vol. 16, No. 1, pp. 22-29, 1990.

[Church et al. 1991] Church, K. W., Gale, W., Hanks, P. and Hindle, D.: "Using statistics in lexical analysis", *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, Zernik U (ed.), Lawrence Erlbaum Associates, pp. 115-164, 1991.

[Ehara et al. 1990] Ehara, T., Ogura, K. and Morimoto, T.: "ATR dialogue database", *Proc. of the 1990 International Conference on Spoken Language Processing*, pp. 1093-1096, 1990.

[Hindle and Rooth 1993] Hindle, D. and Rooth, M.: "Structural ambiguity and lexical relations", *Computational Linguistics*, Vol. 19, No. 1, pp. 103-120, 1993.

[Jelinek 1990] Jelinek, F.: "Self-organized language modeling for speech recognition", *Readings in Speech Recognition*, Waibel, A. and Lee, K. F. (eds.), Morgan Kaufmann Publishers, pp. 450-506, 1990.

[Kita 1992] Kita, K.: *A Study on Language Modeling for Speech Recognition*, Ph.D Thesis, Waseda University, 1992.

[Kita et al. 1993] Kita, K., Ogura, K., Morimoto, T. and Yano, Y.: "Automatically extracting frozen patterns from corpora using cost criteria", *Transactions of Information Processing Society of Japan*, Vol. 34, No. 9, pp. 1937-1943, 1993. (in Japanese)

[Kita et al. 1993b] Kita, K., Hayashi, T. and Yano, Y.: "Corpus-based language learning: Towards practical language learning systems", *Proc. of the 1993 International Conference on Computers in Education*, pp. 355-357, 1993.

[Kupiec 1992] Kupiec, J.: "Robust part-of-speech tagging using a hidden Markov model", *Computer Speech and Language*, No. 6, pp. 225-242, 1992.

[Magerman and Marcus 1990] Magerman, D. M. and Marcus, M. P.: "Parsing a natural language using mutual information statistics", *Proc. of the Eight National Conference on Artificial Intelligence*. pp. 984-989, 1990.

[Nagao 1984] Nagao, M.: "A framework of a mechanical translation between Japanese and English by analogy principle", *Artificial and Human Intelligence*, Elithorn, A. and Banerji, R. (eds.), Elsevier Science Publishers, pp. 173-180, 1984.

[Smadja and McKeown 1990] Smadja, F. A. and McKeown, K. R.: "Automatically extracting and representing collocations for language generation", *Proc. of the 28th Annual Meeting of the Association for Computational Linguistics*, pp. 252-259, 1990.

[Smadja 1991] Smadja, F. A.: "Macrocoding the lexicon with co-occurrence knowledge", *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, Zernik, U. (ed.), Lawrence Erlbaum Associates, pp. 165-189, 1991.

[Smadja 1993] Smadja, F.: "Retrieving collocations from text: Xtract", *Computational Linguistics*, Vol. 19, No. 1, pp. 143-177, 1993.

[Sumita and Iida 1992] Sumita, E. and Iida, H.: "Example-based NLP techniques: A case study of machine translation", *Proc. of the AAAI Workshop on Statistically-Based NLP Techniques*, pp. 90-97, 1992.

[Wolff 1991] Wolff, J. G.: *Towards a Theory of Cognition and Computing.* Ellis Horwood, 1991.

Iterative Alignment of Syntactic Structures for a Bilingual Corpus

Ralph Grishman

Computer Science Department
New York University

Abstract

Alignment of parallel bilingual corpora at the level of syntactic structure holds the promise of being able to discover detailed bilingual structural correspondences automatically. This paper describes a procedure for the alignment of regularized syntactic structures, proceeding bottom-up through the trees. It makes use of information about possible lexical correspondences, from a bilingual dictionary, to generate initial candidate alignments. We consider in particular how much dictionary coverage is needed for the alignment process, and how the alignment can be iteratively improved by having an initial alignment generate additional lexical correspondences for the dictionary, and then using this augmented dictionary for subsequent alignment passes.

Introduction

The process of aligning bilingual corpora can provide valuable information about the source and target languages and about bilingual correspondences. This alignment can be done at several levels. There is already a considerable literature on performing sentence-level alignment and identifying word-level correspondences (for example, [Church 93], [Chen 93], and works cited therein).

Our own work starts with a corpus which has been aligned at the sentence level, and considers the problem of alignment at the level of regularized syntactic structure — a level corresponding approximately to "deep structure" or the F-structure of lexical-functional grammar. Previous studies have shown that at this level, which abstracts

away some of the most apparent surface differences between languages, there is a considerable parallel between language structures [Harris 68, Teller et al. 88].

Alignment at this level serves several purposes. It can be used to identify vocabulary correspondences in a more focused way than sentence-level alignment (thus permitting, for example, identification of lexical correspondences from a single example). It can be used to disambiguate syntactic analyses in one language, using information from the corresponding sentence in the other language [Utsuro et al. 92, Matsumoto et al. 93]. And it can be used to identify correspondences at the level of syntactic case frames and larger syntactic structures, as would be required for a transfer-based machine translation system [Kaji et al. 92, Grishman and Kosaka 92]. The latter has been our principal motivation in developing this alignment procedure.

In the next section, we consider our motivation in somewhat more detail, focusing on the selection of the appropriate level of analysis at which to perform the alignment. The sections which follow describe the alignment algorithm itself, and some of the evaluations which we have made of the algorithm.

Level of Analysis

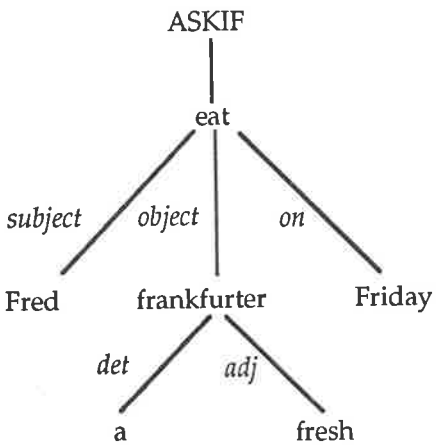
In developing language analysis systems we are always faced with the problem of having to encode large amounts of information. In analyzing text in a single language, for example, we are faced with the problem of capturing selectional constraints — information on the meaningful or allowable combinations of words. In machine translation, using a transfer-based approach, we are faced with the need to specify a large number of rules to map source language into target language structures.

These hurdles are now beginning to be overcome through the use of corpus-based discovery techniques. For monolingual analysis, there have now been several successful efforts at extracting selectional patterns from corpora [Sekine et al. 92, Chang et al. 92, Grishman and Sterling 92]. In the realm of machine translation, there have been two avenues of development. On the one hand, the work on Example-Based Machine Translation [Sato and Nagao 90, Sumita and Iida 91] and on Bilingual Knowledge Bases [Sadler and Vendelman 90] has shown that collections of manually-selected, syntactically analyzed bilingual examples can be an effective source of translation information, substituting for explicitly prepared transfer rules. On the other hand, the

work on Statistically-Based Machine Translation [Brown et al. 90] has shown that bilingual correspondences extracted automatically from corpora, with minimal syntactic processing, can be an effective base for a translation system. Our goal is to combine these two approaches by automatically extracting *structural* correspondences through the alignment of syntactically analyzed and regularized corpora.

Alignment at the word level, as was originally done by IBM, poses difficulties because the correspondences across languages can be quite complex. Using surface syntactic structures, as in some of the EBMT work and the recent work at Hitachi [Kaji et al. 92], simplifies the correspondences and hence the task of alignment. A regularization of the syntactic structures, for example to introduce a single representation of different clausal structures, further improves the correspondence across languages and thus the potential for a discovery procedure to automatically acquire these correspondences from a limited training sample.

In choosing a level of representation, we have sought to perform whatever regularizations can be stated in terms of general syntactic categories. Thus, in English, declarative and interrogative forms, active and passive clauses, relative and reduced relative clauses, are all reduced to a single form. In the resulting form (like f-structure), the basic structure consists of a head and a set of operands in particular syntactic roles, such as *subject, object, indirect object,* etc. for clauses, and *determiner, numeric quantifier, adjectival modifier*, etc. for noun phrases. For example, "Did Fred eat a fresh frankfurter on Friday?" would be represented as



Alignment Algorithm

For our procedure, we assume that the texts have already been aligned at the sentence level, and that both the source and target language texts have been syntactically analyzed and regularized. For the experiments reported here, no additional (selectional) constraints have been applied during parsing, so each source and target sentence will typically have a plurality of parses.

We also assume that we have available a bilingual dictionary which lists typical translations for many of the words in the corpus. We shall consider later just how many are required, but we do not require that the dictionary include translations for all the words, or only the translations used in the corpus. This information might be extracted from a commercial bilingual dictionary, or could itself be derived from a sentence-aligned corpus in an initial stage of processing. We may also have available information about correspondences between role names in the source and target trees.

Given a source and a target sentence tree, an *alignment* is a pairing of a subset of the nodes in the source tree with a subset of the nodes in the target tree. To represent the alignment, we number the nodes in the source tree, $1,...,N_S$, and the nodes in the target tree, $1,...,N_T$; an alignment is then a set of pairs $<s_i,t_i>$, $i=1,...,N_A$, indicating that node s_i of the source tree has been paired with node t_i of the target tree. For an alignment to be well-formed, we require that the relation of dominance in the tree be preserved in the mapping from source nodes to corresponding target nodes; that is, if the alignment includes $<s_i,t_i>$ and $<s_j,t_j>$ and s_i dominates s_j in the source tree, then t_i must dominate t_j in the target tree. (This condition is imposed so that, once correspondences have been identified, the trees can be chopped up into corresponding source and target subtrees.)

The minimal criterion for establishing a correspondence between nodes s_i and t_i is that either

- t_i is a possible translation of s_i as recorded in the bilingual dictionary
- there are one or more pairs $<s_j, t_j>$ in the alignment such that s_i dominates s_j and t_i dominates t_j

- there is a pair in the alignment, $<s_j, t_j>$ such that s_j immediately dominates s_i, t_j immediately dominates t_i, and the role filled by t_i is a possible translation of the role filled by s_i

These minimal criteria would allow for a large number of alternative alignments, so we assign a score to each alignment and select the highest scoring alignment. The score of an alignment is the sum of the scores of the individual correspondences making up the alignment. The score of an individual alignment $<s_i, t_i>$ is based in turn on four terms:

- whether t_i is a possible translation of s_i
- whether s_i dominates any other nodes in the alignment
- the distance from s_i to the other nodes in the alignment which are dominated by s_i (this has a negative weight: nodes which immediately dominate other corresponding nodes are preferred)
- for each node t_j in the alignment which is immediately dominated by t_i, whether the role filled by t_j is a possible translation of the role filled by the corresponding node s_j

The search for alignments proceeds bottom up through the source tree: for each source node, the procedure identifies possible corresponding target nodes, and generates an alignment, or extends previously hypothesized alignments, using each possible correspondence. A form of beam search is employed: a score is associated with each alignment, and only alignments whose score is within some beam width Δ of the score of the best alignment are retained.

When there are multiple parses of the source and target sentence, the alignment procedure is applied between each source parse and each target parse, and selects the source parse and the target parse which together yield the highest-scoring alignment. Unless there are parallel syntactic ambiguities in the source and target sentence, this process can be used to disambiguate (or at least reduce the ambiguity in) the source and target sentences.

Evaluation

For our initial evaluation of this alignment algorithm, we have selected some relatively simple texts: three chapters (73 sentences) from an introductory Spanish textbook, *El Camino Real* [Jarrett and McManus 58], along with English translations of these chapters. One of the byproducts of the alignment process is the selection of a preferred (best-aligning) source language parse, and we have used this as our initial evaluation measure. This is nearly the same measure which has been used in [Matsumoto et al. 93] for the evaluation of their alignment algorithm.

Table 1 shows the improvement in parse accuracy by using the alignment procedure. Without the procedure, the first parse is correct for 43% of the sentences; using the alignment procedure to select a parse yields a correct parse 59% of the time (Table 1, last row).

Method of selecting parse	Percentage of Correct Parses
No alignment	43%
Alignment, using 1/8 of textbook	48%
Alignment, using 1/3 of textbook	52%
Alignment, using entire textbook	59%

Table 1. Parse quality as a function of dictionary size for alignment algorithm.

This first experiment used as a bilingual dictionary the entire dictionary provided with the textbook. To gauge the extent to which successful alignment depended on adequate dictionary coverage, we repeated the alignment procedure using truncated dictionaries, first with 1/3 of the full dictionary, then with 1/8 of the full dictionary. As Table 1 shows, the quality of the alignments correlated with the size of the dictionary.

These experiments indicated the importance of having a procedure which is robust with respect to gaps in the bilingual dictionary. Even the dictionary provided with the textbook did not provide complete coverage, and considerably larger gaps could be

expected when the experiment is extended to use a broad-coverage bilingual dictionary and more complex texts. We therefore implemented an *iterative* alignment algorithm. During one pass through the texts, the procedure collects the correspondences from the best alignment of each sentence. At the end of the pass, it extracts the word correspondences which did not appear in the bilingual dictionary, and adds them to the bilingual dictionary. It also extracts the role correspondences and adds them, along with frequency information, to the table of role correspondences. This extended dictionary and table of role correspondences is then used in the next pass in aligning the text. (Analogous iterative algorithms have been described for *sentence* alignment, in which an initial alignment is used to estimate lexical correspondence probabilities, and these are then used to obtain an improved alignment [Chen 93]).

Through a series of such iterations, the coverage of the bilingual dictionary and table of role correspondences is gradually increased until a limiting state is reached. This is reflected in gradually improving scores on the parsing metric, as shown in Table 2. We began by using only one-eighth of the original dictionary. By the third iteration, the alignments are as good as those obtained with the full original dictionary (no further improvements were obtained by additional iterations).

Iteration Number	Percentage of Correct Parses
1	48%
2	53%
3	59%

Table 2. Improvement of parse quality through iterative alignment.

Discussion

A comparison of our methods with those adopted at Hitachi [Kaji et al. 92] and those adopted at Kyoto and Nara [Utsuro et al. 92, Matsumoto et al. 93] is instructive in understanding some of the alternatives possible in structural alignment.

We noted one difference earlier: the alignment at Hitachi is based on surface structure, whereas our work, and the work at Kyoto and Nara, involves the alignment of "deeper", functional syntactic structures.

There are differences in what constitutes an alignment. Our notion of alignment is consistent with that presented formally in [Matsumoto et al. 93]. For both groups, an alignment is a relation between complete source and target language trees, which respects the dominance relation in the tree (if nodes s_1 and t_1 correspond in the alignment, and so do s_2 and t_2, and s_1 dominates s_2, then t_1 must dominate t_2). In contrast, in Hitachi's approach the alignment of each source tree node to a target tree node is considered independently, and is not directly affected by the alignment of other nodes. (A choice of node alignments, however, may resolve ambiguous word alignments, and therefore indirectly affect subsequent node alignments; as a result, one would expect that in most cases the individual node alignments could be integrated into a tree alignment.)

These differences reflect different goals for the alignment process. The work at Kyoto and Nara has focused on the resolution of syntactic ambiguity. The work at NYU seeks to identify individual structural correspondences within the analysis trees. Both therefore require alignments between tree structures. The Hitachi group, in contrast, builds transfer patterns involving word sequences with limited phrase-structure annotation; these can be constructed by identifying individual correspondences, without aligning entire tree structures.

There are also marked differences in the procedures used to produce the alignments. In the work at Kyoto and Nara, the alignments are built top-down, using a branch-and-bound (backtracking) algorithm to find the best match. The alignment procedure at Hitachi, in contrast, operates bottom-up; it starts by identifying possible word correspondences and then aligns phrases (nodes) of gradually increasing length. It appears that decisions regarding node alignment are made deterministically. This approach fits well with the notion of treating the node alignments independently.

We have chosen to use an alignment algorithm which, like Hitachi's, operates primarily bottom-up. This decision was motivated in part by our earlier studies of parallel bilingual programming language manuals, which indicated that syntactic tree

correspondences were usually very close at the bottom of the tree, for the most sublanguage-specific material, while the trees could diverge considerably at the top (where general vocabulary such as "We will see that ..." was used). Matsumoto et al. note that their procedure encounters some difficulty if the roots of the source and target tree are quite different. In addition, the bottom-up algorithm should be able to handle quite naturally situations where a single source sentence corresponds to multiple target language sentences.

Our choice of a bottom-up algorithm was also motivated in part by considerations of efficiency. The top-down branch-and-bound algorithm can find the optimal match, but because the search space of possible matches is so large, it may take a very long time to do so. Our bottom-up match, guided by the word correspondences and employing limited backtracking (beam search), is not guaranteed to find an optimal alignment, but it appears that it can find acceptable alignments with more limited search. There are, however, cases where the pure bottom-up strategy behaves poorly. This shortcoming is particularly evident in sentences with multiple conjunctions, where a number of low-level alignments will be constructed, most of which will be discarded (due to low scores) when the top levels of the tree are reached (our training texts, while generally syntactically fairly simple, make heavy use of conjunction, presumably because it would be easy for beginning language learners to understand.)

To improve efficiency, we are now experimenting with a combination of top-down and bottom-up search. We begin by proceeding top-down, starting from the root, and continuing so long as there is a close lexical and structural match between source and target trees. When the top-down match stops (because there is some divergence between source and target trees), the remainder of the trees with be matched bottom-up using the procedure previously described.

Application: Transfer Rule Discovery Procedures

As we noted earlier, our objective in creating these alignments is to automatically extract transfer rules from the bilingual corpus. Once an alignment has been created, the source and target trees are "cut" at the nodes in the alignment, producing a set of source tree fragments and target tree fragments. If every node is in the alignment, each tree fragment will be a single level of the tree, indicating how a head plus a set of

syntactic roles in the source language is mapped into a head plus roles in the target language. If the alignment does not include every node, the mapping may go from a single level in the source language to two or more levels in the target language, or vice versa — a "structural transfer". These corresponding tree fragments are then collected and generalized to form the transfer rules of a translation system.

We have completed a rudimentary system of this form for producing translations from Spanish to English. However, because of the simplicity of the sentences in our current training corpus (the first few chapters of our Spanish textbook), almost no structural transfer is needed (once the text is parsed, translation is nearly direct), and so the capability of this approach to acquire and generalize such structural rules is not yet seriously tested. We intend in the near future to extend our training corpus to larger portion of this textbook and to other texts in order to properly gauge the power of our procedure in acquiring structural transfer rules.

Acknowledgment

The work reported in this paper was supported by the National Science Foundation under Grant IRI-9303013.

References

[Brown et al. 90] P. F. Brown, J. Cocke, S. A. DellaPietra, V. J. DellaPietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roosin. A statistical approach to machine translation. *Computational Linguistics* **16** (2), 1990.

[Chang et al. 92] J.-S. Chang, Y.-F. Luo, and K.-Y. Su. GPSM: a generalized probabilistic semantic model for ambiguity resolution. *Proc. 30th Annl. Meeting Assn. for Computational Linguistics*, Newark, DE, 1992, 177-184.

[Chen 93] S. Chen, Aligning sentences in bilingual corpora using lexical information. *Proc. 31st Annl. Meeting Assn. Computational Linguistics,*, Columbus, Ohio, June 1993, 9-16.

[Church 93] K. W. Church, Char_align: a program for aligning parallel texts at the character level. *Proc. 31st Annl. Meeting Assn. Computational Linguistics,*, Columbus, Ohio, June 1993, 1-8.

[Grishman and Kosaka 92] R. Grishman and M. Kosaka. Comparing rationalist and empiricist approaches to machine translation. *Proc. Fourth Int'l Conf. on Theoretical and Methodological Issues in Machine Translation*, Montreal, June 1992.

[Grishman and Sterling 92] R. Grishman and J. Sterling. Acquisition of selectional patterns. *Proc. 14th Int'l Conf. on Computational Linguistics,* Nantes, France, 1992.

[Jarrett and McManus 58] E. M. Jarrett and B. J. M. McManus. *El Camino Real, Book One*. Boston: Houghton Mifflin, 1958.

[Harris 68] Z. Harris, *Mathematical Structures of Language*. New York: Wiley Interscience, 1968.

[Kaji et al. 92] H. Kaji, Y. Kida, and Y. Morimoto. Learning translation templates from bilingual text. *Proc. 14th Int'l Conf. on Computational Linguistics*, Nantes, 1992, 672-678.

[Matsumoto et al. 93] Y. Matsumoto, H. Ishimoto, T. Utsuro, and M. Nagao. Structural matching of parallel texts. *Proc. 31st Annl. Meeting Assn. Computational Linguistics,*, Columbus, Ohio, June 1993, 23-30.

[Sadler and Vendelman 90] V. Sadler and R. Vendelman. Pilot implementation of a bilingual knowledge bank. *Proc. 13th Int'l Conf. on Computational Linguistics*, Helsinki, Finland, 1992, 449-451.

[Sato and Nagao 90] S. Sato and M. Nagao. Toward memory-based translation. *Proc. 13th Int'l Conf. Computational Linguistics,* Helsinki, Finland, 1990, 247-252.

[Sekine et al. 92] S. Sekine, J. Carroll, A. Ananiadou, and J. Tsujii. Automatic learning for semantic collocation. *Proc. Third Conf. Applied Natural Language Processing*, Trento, Italy, 1992, 104-110.

[Sumita and Iida 91] E. Sumita and H. Iida. Experiments and prospects of example-based machine translation. *Proc. 29th Annl. Meeting Assn. for Computational Linguistics*, Berkeley, Ca, 1991.

[Teller et al. 88] V. Teller, M. Kosaka, and R. Grishman. A comparative study of Japanese and English sublanguage patterns. *Proc. Second Int'l Conf. on Theoretical and Methodological Issues in Machine Translation*, Pittsburgh, PA, 1988.

[Utsuro et al. 92] T. Utsuro, Y. Matsumoto, and M. Nagao. Lexical knowledge acquisition from bilingual corpora. *Proc. 14th Int'l Conf. on Computational Linguistics*, Nantes, 1992, 581-587.

Statistical Augmentation of a Chinese Machine-Readable Dictionary

Pascale Fung
Columbia University
Computer Science Department
New York, NY 10027
USA
pascale@cs.columbia.edu

Dekai Wu
HKUST
Department of Computer Science
University of Science & Technology
Clear Water Bay, Hong Kong
dekai@cs.ust.hk

Abstract

We describe a method of using statistically-collected Chinese character groups from a corpus to augment a Chinese dictionary. The method is particularly useful for extracting domain-specific and regional words not readily available in machine-readable dictionaries. Output was evaluated both using human evaluators and against a previously available dictionary. We also evaluated performance improvement in automatic Chinese tokenization. Results show that our method outputs legitimate words, acronymic constructions, idioms, names and titles, as well as technical compounds, many of which were lacking from the original dictionary.

1 Introduction

Finding new lexical entries for Chinese is hampered by a particularly obscure distinction between characters, morphemes, words, and compounds. Even in Indo-European text where words can be separated by spaces, no absolute criteria are known for deciding whether a collocation constitutes a compound word. Chinese defies such distinctions yet more strongly. Characters in Chinese (and in some other Asian languages such as Japanese and Korean) are not separated by spaces to delimit words; nor do characters give morphological hints as

to word boundaries. Each single character carries a meaning and can be ambiguous; most are many-way polysemous or homonymous.

Some characteristics of Chinese words are nonetheless clear. A word in Chinese is usually a bigram (two character word), a unigram, a trigram, or a 4-gram. Function words are often unigrams, and n-grams with $n > 4$ usually are specific idioms. According to the *Frequency Dictionary of Modern Chinese* (FDMC 1986), among the top 9000 most frequent words, 26.7% are unigrams, 69.8% are bigrams, 2.7% are trigrams, 0.007% 4-grams, and 0.0002% 5-grams. Another study (Liu 1987) showed that in general, 75% of Chinese words are bigrams, 14% trigrams, 6% n-grams with $n > 3$.

Inadequate dictionaries have become the major bottleneck to Chinese natural language processing. Broad coverage is even more essential than with Indo-European languages, because not even the most basic lexicosyntactic analysis can proceed without first identifying the word boundaries. Thus a significant number of models for *tokenizing* or *segmenting* Chinese have recently been proposed, using either rule-based or statistical methods (Chiang *et al.* 1992; Lin *et al.* 1992; Chang & Chen 1993; Lin *et al.* 1993; Wu & Tseng 1993; Sproat *et al.* 1994). But all of these approaches rely primarily upon dictionary lookup of the potential segments; in spite of experimental heuristics for handling unknown words in the input text, accuracy is seriously degraded when dictionary entries are missing.

Tokenization problems are aggravated by text in specialized domains. Such documents typically contain a high percentage of technical or regional terms that are not found in the tokenizer's dictionary (machine-readable Chinese dictionaries for specialized domains are not readily available). Most effective tokenizers have domain-specific words added manually to the dictionary. Such manual strategies are too tedious and inefficient in general.

This paper discusses a fully automatic statistical tool that extracts words from an untokenized Chinese text, creating new dictionary entries. In addition, it is desirable to identify regional and domain-specific technical terms that are likely to appear repeatedly in a large corpus. We extended and re-targeted a tool originally designed for extracting English compounds and collocations, Xtract, to find words in Chinese. We call the resulting tool **CXtract**. Words found by CXtract are used to augment our dictionary.

In the following sections, we first describe the modifications in CXtract for finding Chinese words, and the corpus used for training. The resulting words and collocations are evaluated by human evaluators, and recall and precision are measured against the tokens in the training set. The significance of evaluated results will be discussed. Finally, we discuss a preliminary evaluation of the improvement in tokenization performance arising from the

output of our tool.

2 A Collocation Extraction Tool

Xtract was originally developed by Smadja (1993) to extract collocations in an English text. It consists of a package of software tools used to find likely co-occurring word groups by statistical analysis.

In the first stage of Xtract, all frequent bigrams are found. These bigram words are permitted to occur within a window of 10 positions, specifically, at distance between -5 to 5 relative to each other. A threshold is set on the frequency, to discard unreliable bigrams. The remaining bigrams constitute part of the output from Xtract (along with the output from the second stage).

The second stage looks at a tagged corpus and to find collocations of involving more than two words—up to ten—using the bigram words found in the first stage as anchors. Again, a frequency threshold is set to discard unreliable collocations.

Xtract's output consists of two types of collocations. In the simpler case, a collocation is an adjacent word sequence such as "stock market" (extracted from the Wall Street Journal). More general collocations permit flexible distances between two word groups, as in "make a ... decision".

For our purpose, we were interested in looking for adjacent character groups without distances between the groups. We postulated that, just as "stock market" could be regarded as a compound word, we would discover that frequently appearing continuous character groups are likely to be words in Chinese. We were also interested in looking for multi-word collocations in Chinese since these would presumably give us many technical and regional terms.

Because Xtract was originally developed for English, many capabilities for handling non-alphabetic languages were lacking. We extended Xtract to process character-based Chinese texts without tags. Various stages of the software were also modified to deal with untagged texts.

Other parametric modifications arose from the difference between the distribution of characters that make up Chinese words, versus the words that make up English compounds. For example, the frequency threshold for finding reliable bigrams is different because CXtract returns far more Chinese character bigrams than English word bigrams returned by Xtract.

3 Experiment I: Dictionary Augmentation

Our experiments were aimed at determining whether our statistically-generated output contains legitimate words. We are using text from (the Chinese part of) the HKUST English-Chinese Parallel Bilingual Corpus (Wu 1994), specifically, transcriptions of the parliamentary proceedings of the Legislative Council. The transcribed Chinese is formalized literary Cantonese that is closer to Mandarin than conversational Cantonese. However, more vocabulary is preserved from classical literary Chinese than in Mandarin, which affects the ratio of bigrams to other words.

Evaluation of legitimate Chinese words is not trivial. It is straightforward to evaluate those outputs that can be found in a machine-readable dictionary such as the one used by the tokenizer. However, for unknown words, the only evaluation criterion is human judgement. We evaluated the output lexical items from CXtract by both methods.

3.1 Procedure

For Experiment I, we used a portion of the corpus containing about 585,000 untokenized Chinese characters (which turned out to hold about 400 thousand Chinese words after tokenization). The experiment was carried out as follows:

1: A dictionary of all unigrams of characters found in the text was composed. One example is the character 立 (*li*) which can mean "stand" or "establish" by itself.

2: From the unigram list, we found all the bigrams associated with each unigram and obtained a list of all bigrams found.

3: We kept only bigrams which occur significantly more than chance expectation, and which appear in a rigid way (Smadja 1993). This yields a list of possible bigrams and most frequent relative distance between the two characters. The distances are kept between -5 and 5 as in Xtract since this ultimately gives collocations of lengths up to 10, which we found sufficient for Chinese.

4: From this bigram list, we extracted only those bigrams in which the two characters occur adjacently. We assumed such bigrams to be Chinese words. For 立 (*li*), one output bigram was 立法 (*li fa*) which means "legislative", a legitimate word.

5: Using all bigrams (adjacent and non-adjacent) from (3), we extracted words and collocations of lengths greater than two. Outputs with frequency less than 8 were discarded.

6: We divided the output from (5) into lists of trigrams, 4-grams, 5-grams, 6-grams, and m-grams where $m > 6$. One of the trigrams, for example, is 立法局 (*li fa ju*) which means "Legislative Council" and is another legitimate word.

3.2 Results

A portion of the list of bigrams obtained from (4) is shown in Figure 1. We obtained 1695 such bigrams after thresholding.

Part of the output from (5) is shown in Figure 2. The first and the last numbers on each line is the frequency for the occurrence of the n-grams.

Parts of the output from (6) are shown in Figures 3, 4, and 5.

立法	legislative	人士	personage	政府	government
提供	supply	發展	develop(ment)	意見	opinion
部份	partial	包括	include	教育	educate(-tion)
要求	demand	政治	politics	而且	moreover
制度	system	研究	research	一般	in general
市民	citizen	組別	group/category	一個	one *classifier*
法律	law	條例	regulation	報告	report
缺乏	lack/deprivation	關係	relation	如果	if/suppose
修訂	revise(-sion)	表示	express/indicate	關注	attention
任何	any	反映	reaction	一九	nineteen
利益	benefit	部門	department	增加	increase
公平	fair	本身	itself	精神	spirit
工作	job	特別	special	數字	number

Figure 1: Part of the bigram output, with glosses

3.3 Human evaluation

For the first part of the precision evaluation, we relied on human native speakers of Mandarin and Cantonese. Many of the output words, especially domain-specific words and collocations, were not found in the tokenizer dictionary. Most importantly, we are interested in the percentage of output sequences that are legitimate words that can be used to augment the tokenizer.

Four evaluators were instructed to mark whether each entry of the bigram and trigram outputs was a word. The criterion they used was that a word must be able to stand by itself and does not need context to have a meaning. To judge whether 4-gram, 5-gram, 6-gram and m-gram outputs were words, the evaluators were told to consider an entry a word if it

Freq	Collocation
277	..... 副 主 席 先 生 , ...
337	..... 政 府 .....
50	..... 利 益 .....
30	..... 政 府 可 否 告 知 本 局 .....
16	..... 第 五 號 報 告 書 .....
27	..... 明 白 .....
20	..... 私 事 ..... 政 府 當 局 .. 訂 究 證 文 . , . 私 .....
16	..... 謹 此 陳 辭 , 支 持 動 議 。 ..... 議 員 致 辭 .....
12	..... 工 商 界 .....

Figure 2: Part of the CXtract output

不 公 平	不 知 交 和	供 某 人 政 a21
中 小 學	中 英 雙 方	兩 個 市 政 局
公 務 員	中 國 政 府	券 及 期 貨 事
及 他 方	公 共 援 助	委 員 會 提 出
及 其 他	分 區 直 選	的 生 活 方 式
尤 其 是	支 持 動 議	的 投 票 制 度
大 多 數	文 件 所 載	的 選 舉 制 度
大 故 是	人 權 法 案	的 選 舉 制 度
大 家 都	夾 心 階 層	建 議 修 訂 內
大 部 份	和 會 計 師	很 大 的 缺 失
工 市 政	的 代 表 性	施 政 報 告 中
立 法 局	的 是 人 一	施 政 報 告 內
任 何 人	直 選 議 席	副 主 席 先 生
全 日 制	社 會 人 士	商 務 委 員 會
工 商 界	社 會 福 利	專 責 委 員 會

Figure 3: Part of the trigram, 4-gram and 5-gram output

was a sequence of shorter words that taken together held a conventional meaning, and did not require any additional characters to complete its meaning.

Besides *correct*, the evaluators were given three other categories to place the n-grams. *Wrong* means the entry had no meaning or an incomplete meaning. *Unsure* means the evaluator was unsure. Note that the percentage in this category is not insignificant, indicating the difficulty of defining Chinese word boundaries even by native speakers. *Punctuation* means one or more of the characters was punctuation or ASCII markup.

Tables 1 and 2 show the results of the human evaluations. The *Precision* column gives

一九九七年後	after the year 1997
人權法案條例	Human Rights Bill
中英聯合聲明	Sino-British Joint Declaration
以及他們所提	and as they mentioned
加上九七年能	additionally in 1997 we can
本人謹此陳辭	I hereby move that
刑事罪行條例	Criminal Law Bill
至一九九七年	until the year 1997
我們必須審刻	we must examine
我們應該予支	we should allocate
我們應該糾支	we should correct
見是民上是內	*error*
兩個市政局的	two Urban Council's
券及期貨事務	*error* and Commodity Affairs
動議投贊成票	move to cast a supporting vote
第五號報告書	The Number 5 Report
港特別行政區	Hong Kong Special Administrative Region
機場核心工程	Airport Core Project
選舉委員會的	of the Elective Committee
總督施政報告	The Executive Report of the Governor
謹此提出動議	hereby beg to move

Figure 4: Part of the 6-gram output, with glosses

the percentage correct over total n-grams in that category.

We found some discrepancies between evaluators on the evaluation of *correct* and *unsure* categories. Most of these cases arose when an n-gram included the possessive 的 (*de*), or the copula 是 (*shi*). We also found some disagreement between evaluators from mainland China and those from Hong Kong, particular in recognizing literary idioms.

The average precision of the bigram output was 78.13%. The average trigram precision was 31.3%; 4-gram precision 36.75%; 5-gram precision 49.7%; 6-gram precision 55.2%; and the average m-gram precision was 54.09%.

3.4 Dictionary/text evaluation

The second part of the evaluation was to compare our output words with the words actually present in the text. This gives the recall and precision of our output with respect to the training corpus. Unfortunately, the training corpus is untokenized and too large to tokenize by hand. We therefore estimated the words in the training corpus by passing it through an automatic tokenizer based on the BDC dictionary (BDC 1992). Note that this dictionary's

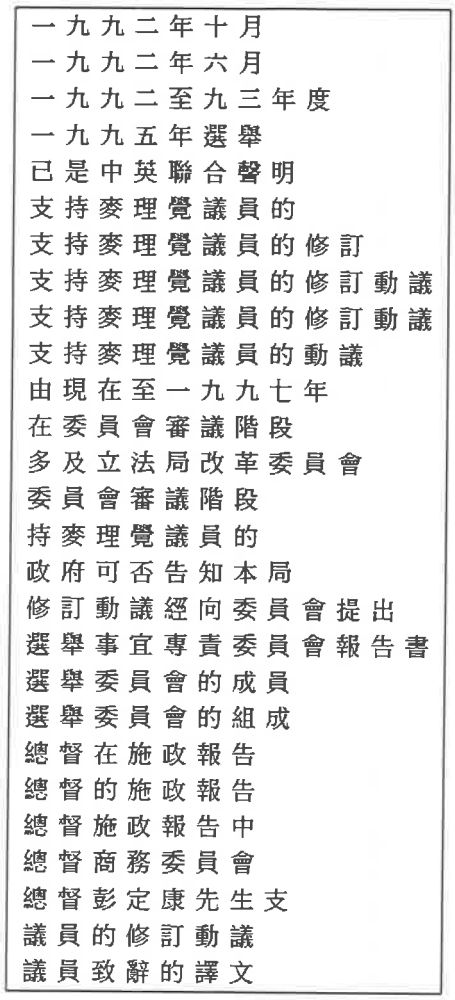


Figure 5: Part of the m-gram output

entries were not derived from material related to our corpus. The tokens in the original tokenized text were again sorted into unique bigrams, trigrams, 4-grams, 5-grams, 6-grams, and m-grams with $m > 6$. Table 3 summarizes the precision, recall, and augmentation of our output compared to the words in the text as determined by the automatic tokenizer. *Precision* is the percentage of sequences found by CXtract that were actually words in the text. *Recall* is the percentage of words in the text that were actually found by CXtract. *Augmentation* is the percentage of new words found by CXtract that were judged to be correct by human evaluators but were not in the dictionary.

The recall is low because CXtract does not include n-grams with frequency lower than 8. However, we obtained 467 legitimate words or collocations to be added to the dictionary

Table 1: Human Evaluation of the Bigram Output Precision

Evaluator	wrong	unsure	punctuation	precision
A	339	53	111	
	20%	3.1%	6.5%	75.2%
B	264	31	111	
	15.6%	1.8%	6.5%	81.4%
C	269	118	111	
	15.87%	6.96%	6.5%	75.6%
D	289	23	111	
	17%	1.4%	6.5%	80.3%

Table 2: Human Evaluation of n-gram Output Precision

Evaluator	n	wrong	correct	unsure	punctuation	precision
A	3	205	81	33	25	23.5%
	4	98	89	5	20	44.1%
	5	33	48	1	6	54.5%
	6	9	32	5	1	68%
	m	14	32	3	0	65.3%
D	3	296	101	23	25	29.4%
	4	102	75	2	23	37.13%
	5	36	44	2	6	50%
	6	20	26	0	1	55.3%
	m	18	27	0	4	55.1%
E	3	168	134	16	26	39%
	4	89	81	10	22	40.1%
	5	29	44	5	10	50%
	6	10	0	11	1	53.2%
	m	12	0	7	4	53.06%
C	3	210	112	0	22	32.6%
	4	131	52	0	19	25.7%
	5	40	39	0	9	44.3%
	6	25	21	0	1	44.7%
	m	24	21	0	4	42.9%

and the total augmentation is 5.73%. The overall precision is 59.3%.

However, we believe the frequency threshold of 8 was too low and the 585K character size of the corpus was too small. Most of the "garbage" output had low frequencies. The precision rate can be improved by using a larger data base and raising the threshold as in

Table 3: Precision, Recall and Augmentation of CXtract Output

n	Token types	CXtract	Precision	Recall	Augmentation
2	6475	1201	852 (70.9%)	662 (10.2%)	190 (2.9%)
3	721	344	115 (33.4%)	10 (1.4%)	105 (14.6%)
4	911	202	75 (37.1%)	7 (0.008%)	68 (7.5%)
5	38	88	43 (48.9%)	0 (0%)	43 (113.2%)
6	7	47	29 (61.7%)	0 (0%)	29 (414.2%)
m	4	49	32 (65.3%)	0 (0%)	32 (800%)
Total	8156	1931	1146 (59.3%)	769 (14%)	467 (5.73%)

Experiment II.

In the following sections, we discuss the significance of the evaluated results.

3.5 Bigrams are mostly words

Using human evaluation, we found that 78% of the bigrams extracted by our tool were legitimate words (as compared with $70.9\% + 2.9\% = 73.8\%$ by evaluation against the automatic tokenizer's output). Of all n-gram classes, the evaluators were least unsure of correctness for bigrams, although quite a few classical Chinese terms were difficult for some of the evaluators.

Since the corpus is an official transcript of formal debates, we find many terms from classical Chinese which are not in the machine-readable dictionary, such as 謹此 (*jin ci*, "I hereby").

Some of the bigrams are acronymic abbreviations of longer terms that are also domain specific and not generally found in a dictionary. For example, 中英 (*zhong ying*) is derived from 中國,英國 (*zhong guo, ying guo*), meaning Sino-British. This acronymic derivation process is highly productive in Chinese.

3.6 The whole is greater than the sum of parts

What is a *legitimate* word in Chinese? To the average Chinese reader, it has to do with the vocabulary and usage patterns s/he acquired. It is sometimes disputable whether 立法局 (*li fa ju*, "Legislative Council") constitutes one word or two. But for the purposes of a machine translation system, for example, the word 局 (*ju*) may be individually translated not only into "Council" but also "Station", as in 警察局 (*jing cha ju*, "Police Station"). So we might incorrectly get "Legislative Station". On the other hand, 立法局 (*li fa ju*) as a

single lexical item always maps to "Legislative Council"

Another example is 大部份 (*da bu fen*) which means "the majority". Our dictionary omits this and the resulting tokenization is 大 (*da*, "big") and 部份 (*bu fen*, "part/partial"). It is clear that "majority" is a better translation than "big part".

3.7 Domain specific compounds

Many of the n-grams for $n > 3$ found by CXtract are domain-specific compounds. For example, due to the topics of discussion in the proceedings, "the year 1997" appears very frequently.

Longer terms are frequently abbreviated into words of three or more characters. For example, 中英雙方 (*zhong ying shuang fang*) means "bilateral Sino-British", and 中英聯合聲明 (*zhong ying lian he sheng ming*) means "Sino-British Joint Declaration". Various titles, committee names, council names, projects, treaties, and joint-declarations are also found by our tool. Examples are shown in Figure 6.

Although many of the technical terms are a collocation of different words and sometimes acceptable word boundaries are found by the tokenizer, it is preferable that these terms be treated as single lexical items for purposes of machine translation, information retrieval, or spoken language processing.

3.8 Idioms and *cheng yu*

From n-gram output where $n > 3$, we find many idiomatic constructions that could be tokenized into series of shorter words. In Chinese especially, there are many four character words which form a special idiomatic class known as 成語 (*cheng yu*). There are dictionaries of *cheng yu* with all or nearly all entries being four character idioms (e.g., Chen & Chen 1983). In the training corpus we used, we discovered new *cheng yu* that were invented to describe a new concept. For example, 夾心階層 (*jia xin jie ceng*) means "sandwich class" and is a metaphorical term for families who are not well off but with income just barely too high to qualify for welfare assistance. Such invented terms are highly domain dependent, as are the usage frequencies of established *cheng yu*.

3.9 Names

Tokenizing Chinese names is a difficult task (Sproat *et al.* 1994) because Chinese names start with a unigram or bigram family name, and are followed by a given name freely composed of one or two characters. The given name usually holds some meaning, making it hard

白皮書	White Paper
行政局	Executive Council
工商界	Industry and Trade
立法局	Legislative Council
保安司	Security Secretary
財政司	Financial Secretary
經濟司	Economics Secretary
聯合聲明	Joint Declaration
一九九七年	the year 1997
立法局選舉	Election of the Urban Council
商務委員會	Commerce and Trading Committees
專責委員會	Select Committees
選舉委員會	Elective Committees
醫院管理局	Hospital Administration Committee
警務處處長	Police Chief
人權法案條例	Human Rights Bill
中英聯合聲明	Sino-British Joint Declaration
刑事罪行條例	Criminal Law Bill
多議席單票制	many-seats one-vote system
港特別行政區	Hong Kong Special Administrative Region
機場核心工程	Airport Core Project
計由現在至一九九七年	counting from now until the year 1997
由現在至一九九七年	from now until the year 1997
委員會審議階段	examining period of the committee
教育統籌司員會	Education Commission
總督商務委員會	Trading Committee of the Governor

Figure 6: Some domain specific terms found by CXtract, with glosses

to distinguish names from other words. For names, we do not want to tokenize them into separate characters. In a large corpus, names are often frequently repeated. For example, in our data, the names of some parliamentary members are extracted by our tool as separate lexical items. Examples are shown in Figure 7. The last two characters of each example are the person's title.

4 Experiment II: Tokenization Improvement

Given the significant percentage of augmented words in Experiment I, we can see that many entries could be added to the dictionary used for automatic tokenization.

In the next stage of our work, we used a larger portion of the corpus to obtain more

Chinese words and collocations, and with higher reliability. These items were converted into dictionary format along with their frequency information.

To obtain a baseline performance, the tokenizer was tested with the original dictionary on two separate test sets. It was then tested with the statistically-augmented dictionary on the same test sets. Each of the tokenization outputs was evaluated by three human evaluators.

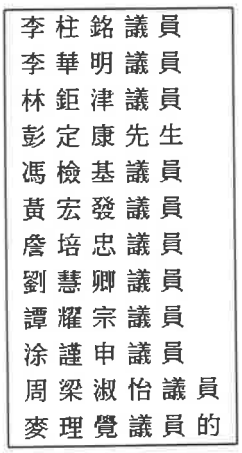


Figure 7: Some names and titles found by CXtract

4.1 Procedure

As training data we used about 2 million Chinese characters taken from the same HKUST corpus. This is about 4 times the size used in Experiment I. The tokenizer we used employs a maximal matching strategy with frequency preferences.

The original dictionary for the tokenizer holds 104,501 entries and lacks many of the domain-specific and regional words found in the corpus.

From the first stage of CXtract, we obtained 4,196 unique adjacent bigrams. From the second stage, we filtered out any CXtract output that occurred less than 11 times and obtained 7,121 lexical candidates. Additional filtering constraints on high-frequency characters were also imposed on all candidates.[1] After all automatic filtering, we were left with 5,554 new dictionary entries.

Since the original dictionary entries employed frequency categories of integer value from 1 to 5, we converted the frequency for each lexical item from the second stage output to

[1] A refined version of the linguistic filtering is discussed in Wu & Fung (1994).

this same range by scaling. The adjacent bigrams from the first stage were assigned the frequency number 1 (the lowest priority).

The converted CXtract outputs with frequency information were appended to the dictionary. Some of the appended items were already in the dictionary. In this case, the tokenization process uses the higher frequency between the original dictionary entry and the the CXtract-generated entry.

The total number of entries in the augmented dictionary is 110,055, an increase of 5.3% over the original dictionary size of 104,501.

4.2 Results

Two independent test sets of sentences were drawn from the corpus by random sampling with replacement. TESTSET I contained 300 sentences, and TESTSET II contained 200 sentences. Both sets contain unretouched sentences with occasional noise and a large proportion of unknown words, i.e., words not present in the original dictionary. (Sentences in the corpus are heuristically determined.)

Each test set was tokenized twice. *Baseline* is the tokenization produced using the original dictionary only. *Augmented* is the tokenization produced using the dictionary augmented by CXtract.

Three human evaluators evaluated each of the test sets on both baseline and augmented tokenizations. Two types of errors were counted: false joins and false breaks. A false join occurs where there should have been a boundary between the characters, and a false break occurs where the characters should have been linked. A conservative evaluation method was used, where the evaluators were told to not to mark errors when they felt that multiple tokenization alternatives were acceptable.

The results are shown in Tables 4, 5, and 6. Baseline error is computed as the ratio of the number of errors in the baseline tokenization to the total number of tokens found. Augmented error is the ratio of the total number of errors in the augmented tokenization to the total number of tokens found.

Our baseline rates demonstrate how sensitive tokenization performance is to dictionary coverage. The accuracy rate of 76% is extremely low compared with other reported percentages which generally fall around the 90's (Chiang *et al.* 1992; Lin *et al.* 1992; Chang & Chen 1993; Lin *et al.* 1993). We believe that this reflects the tailoring of dictionaries to the particular domains and genres on which tokenization accuracies are reported. Our experiment, on the other hand, reflects a more realistic situation where the dictionary and

Table 4: Result of TESTSET I - 300 sentences

Evaluator	# tokens	Baseline # errors	Error rate	Accuracy	# tokens	Augmented # errors	Error rate	Accuracy
A	4194	1128	27%	73%	3893	731	19%	81%
F	4194	1145	27%	73%	3893	713	18%	82%
G	4194	1202	29%	71%	3893	702	18%	82%

Table 5: Result of TESTSET II - 200 sentences

Evaluator	# tokens	Baseline # errors	Error rate	Accuracy	# tokens	Augmented # errors	Error rate	Accuracy
A	3083	737	24%	76%	2890	375	13%	87%
H	3083	489	16%	84%	2890	322	11%	89%
I	3083	545	18%	82%	2890	339	12%	88%

Table 6: Average accuracy and error rate over all evaluators and test sets

Experiment	Total # tokens	Average error	Error rate	Accuracy
Baseline	7277	1749	24%	76%
Augmented	6783	1061	16%	84%

text are derived from completely independent sources, leading to a very high proportion of missing words. Under these realistic conditions, CXtract has shown enormous utility. The error reduction rate of 33% was far beyond our initial expectations.

5 Conclusion

We have presented a statistical tool, CXtract, that identifies words without supervision on untagged Chinese text. Many domain-specific and regional words, names, titles, compounds, and idioms that were not found in our machine-readable dictionary were automatically extracted by our tool. These lexical items were used to augment the dictionary and to improve tokenization.

The output was evaluated both by human evaluators and by comparison against dictionary entries. We have also shown that the output of our tool helped improve a Chinese tokenizer performance from 76% to 84%, with an error reduction rate of 33%.

6 Acknowledgement

We would like to thank Kathleen McKeown for her support and advice, and Frank Smadja and Chilin Shih for helpful pointers. We would also like to thank our evaluators, Philip Chan, Eva Fong, Duanyang Guo, Zhe Li, Cindy Ng, Derek Ngok, Xuanyin Xia, and Michelle Zhou. The machine-readable dictionary (BDC 1992) was provided by Behavior Design Corporation.

References

BDC. 1992. *The BDC Chinese-English electronic dictionary (version 2.0)*. Behavior Design Corporation.

CHANG, CHAO-HUANG & CHENG-DER CHEN. 1993. HMM-based part-of-speech tagging for Chinese corpora. In *Proceedings of the Workshop on Very Large Corpora*, 40–47, Columbus, Ohio.

CHEN, YONG-ZHEN & SPRING CHEN. 1983. *Chinese idioms and their English equivalents*. Hong Kong: Shang Wu Yin Shu Ju.

CHIANG, TUNG-HUI, JING-SHIN CHANG, MING-YU LIN, & KEH-YIH SU. 1992. Statistical models for word segmentation and unknown resolution. In *Proceedings of ROCLING-92*, 121–146.

FDMC. 1986. *Xiandai hanyu pinlu cidian (Frequency dictionary of modern Chinese)*. Beijing Language Institute Press.

LIN, MING-YU, TUNG-HUI CHIANG, & KEH-YIH SU. 1993. A preliminary study on unknown word problem in Chinese word segmentation. In *Proceedings of ROCLING-93*, 119–141.

LIN, YI-CHUNG, TUNG-HUI CHIANG, & KEH-YIH SU. 1992. Discrimination oriented probabilistic tagging. In *Proceedings of ROCLING-92*, 85–96.

LIU, Y. 1987. New advances in computers and natural language processing in China. *Information Science*, 8:64–70. In Chinese.

SMADJA, FRANK. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.

SPROAT, RICHARD, CHILIN SHIH, WILLIAM GALE, & N. CHANG. 1994. A stochastic word segmentation algorithm for a Mandarin text-to-speech system. In *Proceedings of the*

32nd Annual Conference of the Association for Computational Linguistics, Las Cruces, New Mexico. To appear.

WU, DEKAI. 1994. Aligning a parallel English-Chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd Annual Conference of the Association for Computational Linguistics*, Las Cruces, New Mexico. To appear.

WU, DEKAI & PASCALE FUNG, 1994. Improving Chinese tokenization with linguistic filters on statistical lexical acquisition. Submitted.

WU, ZIMIN & GWYNETH TSENG. 1993. Chinese text segmentation for text retrieval: Achievements and problems. *Journal of The American Society for Information Science*, 44(9):532–542.

Comparing the Retrieval Performance of English and Japanese Text Databases

Hideo Fujii and W. Bruce Croft

Computer Science Department
University of Massachusetts, Amherst, MA 01003
e-mail: fujii@cs.umass.edu croft@cs.umass.edu

Abstract

The retrieval effectiveness for English and Japanese full-text databases are studied using the INQUERY retrieval system. Two series of experiments - short queries and longer TIPSTER queries - were examined. For short queries, Japanese generally performed more effectively than English. For longer queries, relative effectiveness showed little correlation among various query strategies. This result suggests that the best Japanese query processing strategy may be quite different from the English one.

1. Introduction - The Problem of Language Comparison in the Text Retrieval

Text retrieval systems provide a good test-bed for language processing technologies. Any qualitative or quantitative aspects of the language, i.e., lexicon, morphology, syntax, semantics and pragmatics, can be applied to these systems. A query as a representation of the user's *information need*, is entered to a retrieval system, and the system retrieves the *relevant* documents from the (possibly gigabytes of) full-text database. Information retrieval (IR) relies on using the linguistic and statistical characteristics of the text. A comparative study between two languages may help to improve our understanding of this process.

The question is how and which effective retrieval techniques for one language can be transferred into another language. That is, our ultimate goal is to discover a *universal strategy* for retrieval across various languages. Here, we examine the retrieval effectiveness for English and Japanese as an example.

Various language-dependent modules are used in an IR system. For example, the algorithm for English word stemming depends on morphological knowledge of the language. Linguistic aspects may significantly affect the retrieval effectiveness as measured by, for example, *recall* (i.e., proportion of relevant documents retrieved in response to the query) and *precision* (i.e., proportion of retrieved documents that are relevant). A good retrieval system will show high performance in both measurements. To achieve this, we must construct a suitable structure of language dependent modules, and design the retrieval strategy to maximize the utilization of the statistical and linguistic characteristics of the text.

Japanese has many different characteristics from English, such as lexicon (e.g., number of loan words), morphology (e.g., no plural form), syntax (e.g., S-O-V word order), pragmatics (e.g., the paragraph structure is less well defined), and written language system (e.g., Chinese characters and no spaces between words).

2. INQUERY Retrieval System

As the basis for our retrieval experiments, we used the INQUERY retrieval system. INQUERY is a probabilistic retrieval system based on a Bayesian inference network [Turtle, 1991, Callan, Croft & Harding, 1992]. In response to a query, it produces an ordered document list based on the estimated conditional probability of satisfying the user's information need. INQUERY itself is language-independent, because it assumes only very general statistical properties such as "term importance is proportional to the ... frequency of each term ... and inversely proportional to the total number of documents to which each term is assigned" [Salton & McGill, 1983].

The INQUERY system works as a core retrieval engine, and various language dependent modules are added to this core. In our previous research, INQUERY demonstrated good retrieval effectiveness for both English [Turtle, 1991; Turtle & Croft, 1991] and Japanese [Fujii & Croft, 1993].

In the English version of INQUERY, a stemming routine, a stopword list, special proper noun recognizers, etc. were implemented as the language-dependent components [Callan, Croft & Harding, 1992].

For Japanese, two indexing methods were previously studied [Fujii & Croft, 1993], namely the *word-based* and *character-based* methods. The word-based method extracts words as in English, whereas the character-based method uses single Kanji characters as indexing units. We reported that the character-based approach can achieve better retrieval effectiveness than the traditional word-based system. A third method, *mixed-mode*, is proposed in this paper.

For word-based indexing in Japanese, each word must be segmented separately. To solve this problem, a program called JUMAN [Matsumoto, Kurohashi, Myoki, et al., 1991] was used to segment documents and queries. The character-based indexing does not have this problem since it discards all inflectional Kana, and uses every Kanji.

In INQUERY, a query can be structured with several retrieval operators, such as *phrase* or *proximity* as well as (probabilistic versions of) the usual Boolean operators, to improve the retrieval effectiveness. A natural language query is translated into this form of structured query using a simple language processing technology, then each operator forms an

intermediate node in the inference network. The query formulation strategy indicates how to combine operators. The major operators used in our experiments are shown in Table 1. We will see several examples in the next two sections.

Table 1. Major INQUERY operators.

[Operator]:	[Action]
#and, #or, #not:	probabilistic version of Boolean operations
#sum:	returns the mean of argument beliefs
#wsum:	returns the weighted mean of argument beliefs
#max:	returns the maximum of argument beliefs
#*n*(proximity):	every adjacent arguments must occur, in order within distance *n*
#uw*n*:	similar to #*n*, except all terms must occur, in any order, in a size *n* window
#phrase:	applies #3 when the phrase occurs frequently, otherwise it applies #sum
#syn:	arguments are considered as synonyms

3. Experiments with Short Queries

We classified our experiments into two types: 1) short queries, and 2) long TIPSTER queries. These are expected to behave differently since a long query contains more structural patterns such as syntactical structure.

In this section, we discuss experiments with short queries including: 1) a general performance comparison; 2) a test for the effects of various retrieval operators; 3) a test for the effects of word distance; 4) the performance differences from various indexing methods for Japanese. Before discussing the results, we describe the test collections used in the experiments.

3.1 Test Collections

There are various test collections in English [Frake and Baeza-Yates, 1993], but currently, there is no standard collection for Japanese. Although an effort to develop a Japanese standard test collection for IR is under way [Kimoto, Tanaka, Ishikawa, et al., 1993], it is not currently available, and is not designed for multi-lingual comparative study.

Before describing the procedure to create our test collections, let us briefly consider the meaning of language comparative collections. Some experienced database searchers may have intuitions about whether English or Japanese is easier for accessing the desired documents. But, how can we justify this intuitive knowledge? Clearly, we need to control the experimental conditions for the comparison. For this, translated texts may be ideal. This is still, however, a questionable method because the translation is obviously conditioned by other language structures such as the selection of the translated vocabulary, syntactical structure,

the paragraph structure, the text style, etc.

Our procedure for constructing Japanese and English test collections is in Appendix 1. Although it is still far from ideal, there are substantial uniformities in our collections - the subject, style, text length, etc. Table 2 shows the summary of our collections and queries in this experiment.

Table 2. Summary of the test collections.

Collection	< English >	< Japanese >
form:	newspaper articles	newspaper articles
subject:	joint ventures in business	joint ventures in business
source:	Wall-Street Journal 1987-91	Mostly from Nikkei-Shinbun 1987-91
collection size:	890 articles (1255 KB)	890 articles (972 KB)
article length:	mean: 1192.0B	mean: 945.8B (*1.25=1188.8)
	S.D.: 732.3B	S.D.: 580.3B (*1.25=725.4)
	min/max: 172/4922B	min/max: 138/4044B
Queries		
# of queries:	25 (translated from Japanese)	25
query size:	5.2 words/query	8.7 chars/query

3.2 Queries - Phrase Structures

A short query is generally expressed as a sentence containing several keywords. The keywords are translated into an intermediate structured form according to the phrase structure. For example, a query, "I want to know about the advancement of Japanese companies in southeastern Asia" could be translated into "#sum(advancement #phrase(Japanese companies) #phrase(southeastern Asia))".

There are four models for short queries, namely NLQ, SHORT, LONG, and JOINED. The NLQ (=natural language query) model does not assume any structure between keywords. The SHORT model groups a set of Kanji characters in a word (or a compound). A Kanji character roughly corresponds to a morpheme. The LONG model clusters nouns (with adjective modifications in English), e.g., Tounan [southeast(*n.*)] Ajia [Asia] [= Southeastern Asia]. The JOIN model puts together LONG phrases which are connected by "-no" [of] in Japanese, "of" or "in" in English. The insight here is that such conjunctions indicate strong connections and often can be transformed into a single noun compound. For example, "Nihon [Japan(*n.*)] no [of] Kigyou [company(*n.*)]" becomes "Nihon-kigyou" in Japanese, or "Japanese(*adj.*) language" for "language of Japan", or "business(*n.*) people" for "people in a business" in English. JOIN is a conservative expansion of LONG without using an arbitrary prepositional phrase.

Please see the detailed discussion in Fujii & Croft, 1993. Figure 1 gives examples.

```
<< English >>
Original Form:"advancement of Japanese companies into southeastern Asia"
NLQ:    #sum( advancement of Japanese companies into southeastern Asia )
LONG:   #sum( advancement of #phrase( Japanese companies ) into #phrase(
southeastern Asia) )
JOINED:#sum( #phrase(advancement of Japanese companies) into #phrase(
southeastern Asia) )

<< Japanese >>
Original form: " 日本企業の東南アジア進出 "
                 ----- ----- --- ----- --------- -----
                [Japan] [company] [of] [southeast] [Asia]  [advancement]

NLQ:    #sum( 日 本 企 業 東 南 アジア 進 出 )
SHORT:  #sum( #phrase(日本) #phrase(企業) #phrase(東南) アジア
        #phrase(進出) )
LONG:   #sum( #phrase(日本 企業) #phrase(東南 アジア 進出) )
JOINED: #phrase( 日本 企業 東南 アジア 進出 )
```

Figure 1. Example of English/Japanese queries.

3.3. General Comparison of English and Japanese Retrieval Performance

Figure 2 shows the recall-precision curves of the two languages. Japanese texts performed better than the English at all recall levels, especially at low recall. Japanese showed 34% higher precision than English in average (27.8 vs. 37.3), and at the low-end of recall, it was 67% higher (42.2 vs. 70.3). Our test collections seem to be appropriately organized since the precision at 100% recall of both languages is almost the equal.

There are two possible factors to explain this effectiveness - *lexical ambiguity*, and *synonymy*. We should determine how these factors work in the mechanism of retrieval.

By lexical ambiguity (e.g., homonyms, polysemy, meaning inclusion, zero morphology, etc.), a word may carry more than one meaning. A less ambiguous query can specify more exactly the concepts that the person wants to express. Lexical ambiguity is related to the precision of retrieval because of the amount of noise.

By synonymy, a concept could be represented by more than one lexical or phrasal entities. To retrieve documents described in different synonymous terms, the query should list those synonyms to include such variations. Synonyms are related to recall.

Our experiments suggest that, for Japanese, lexical ambiguity is the dominant factor for determining the general retrieval performance of the language.

One possible explanation for the less ambiguous nature of Japanese is that Kanji words,

which are Chinese origin, have more specific meaning than native Japanese words which are often written in Hiragana [Matsuo, Nishio & Tanaka, 1965], and they are preferably used as a formal expression in a written text. This explanation may be generally extended to other loan words such as Katakana words which came mostly from English. For example, *mishin* is a Japanese word which is a phonetic translation of "machine", but it is used specially for the sewing machine (as by meaning inclusion). Thus, Japanese lexical semantics is more narrowly specified than in English.

Although data is not shown here, both languages showed no significant improvements in LONG and JOINED using the #phrase operator. The phrase in the INQUERY is a *statistical* operator, but not *linguistic*. We may need to put more linguistic constraints into phrase handling.

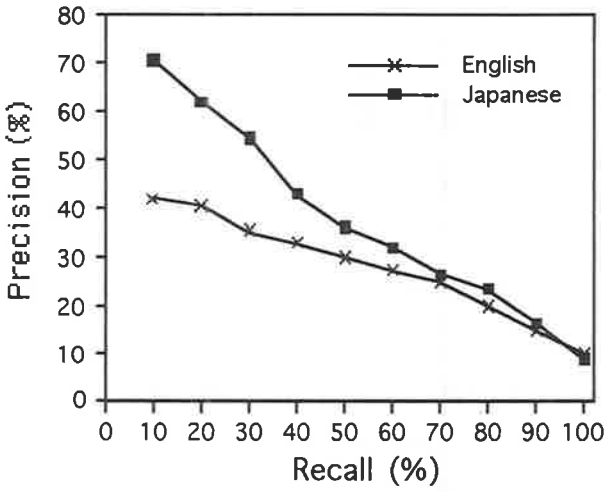


Figure 2. P-R Curve for English and Japanese.
(25 NLQ queries, using #phrase; Word-based in Japanese)

3.4 Performance Differences for Various Operators

This experiment shows a *relative comparison* between two languages unlike the first experiment. *#Phrase, #max, #and,* and *#3* are tested. Table 3 is the result.

Although these results didn't show better performance than NLQ for common phrases (rather than idiomatic phrases, e.g., "White House" which will obviously perform better with a phrase operator), #phrase worked best in both languages, and the correlation coefficient was very high (=0.99).

Table 3. Operator differences of effectiveness.

Operator	#phrase	#max	#and	#3(prox)
Japanese	36.8	35.3	33.9	18.6
English	27.8	26.6	23.1	10.2

3.5 Effect of Word Distance

This experiment is also a relative comparison between the two languages. It shows the effect of the word distance of the *proximity* operator (Figure 3). Both languages performed in a similar way - increasing effectiveness with window size. The slower increase in Japanese suggests the strong locality of word distribution in the text. One explanation is as follows.

Kajiwara [1993] pointed out that newer Kanji words are less likely to be used in a compound, and also it evolves more semantically applicable form. In the *modular morphology* [Kageyama, 1989], the word formation is divided into the lexical units (*type-A* in his term. We call this *lexical word*) and syntactical units (*type-B*. Here, *syntactic word*) under the certain morphological constraints. So, in the process, syntactical Kanji words could be naturally selected rather from lexical units of morphology. Lexical words are semantically opaque, and syntactical vocabulary are transparent and more morphologically productive.

Thus, if two concepts of syntactical words have cooccurred in a sentence, they will easily produce a compound. In contrast, lexical words can be less constrained in their placement in sentences or beyond them. If this hypothetical mechanism is correct, we can take two distinct search strategies for Japanese syntactic words (e.g., many common Kanji words) and lexical words (e.g., neologism, Katakana words, etc.).

As a consequence of above conjecture, we predict that syntactical approach (of a sentence) in Japanese will be more effective than in English. This is a theme of our research.

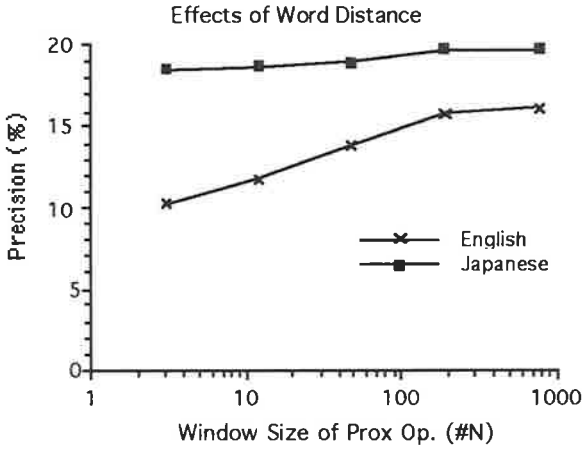


Figure 3. Effects of the word distance.

3.6 Performance Differences of Indexing Methods in Japanese

This experiment shows how a writing system affects the retrieval performance. Japanese language has a very characteristic usage of Kanji words, which are loan words from Chinese since the early age of Japanese written language development. Lots of them (especially

for the abstract concepts) are formed as two-character words. Since the Kanji character is an ideogram and it is nearly equivalent to the morpheme, there is a way to use each Kanji character instead of a word as an indexing unit. Also we developed a method to index both character level and word level at the same time - called *mixed-mode*. Figure 4 shows the result of three indexing methods.

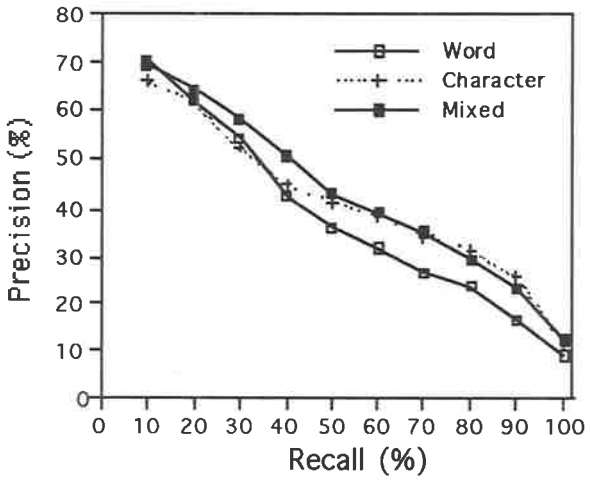


Figure 4. Effect of three indexing methods.

We already reported that the character-based method potentially performs better than word-based [Fujii & Croft, 1993]. In our new results, we found following three results: 1) new data again supported the above point in general (9% better than word-based), 2) the gain of the character-based performance was mostly at the middle to high recall level - a *thesaurus effect* of ideographic characters [Fujii & Croft, 1993], and 3) more importantly, the mixed-mode performed best at most levels (14% improvement in average).

4. Experiments with a Long TIPSTER Queries

A TIPSTER query [Harman, 1992] is structured as a *topic*, which mainly contains: 1) title(<title>), 2) description(<Desc>), 3) narrative(<Narr>), 4) concepts(<Con>), and 5) factors (<Fac>). The <Desc> and <Narr> are natural language descriptions - <Desc> is a description of <title>, and <Narr> is a more detail explanation, for example the criteria of relevant judgment. <Con> is a set of keyword groups. A query example is shown in Appendix 2.

Various query formulation techniques are examined using the topic of German joint ventures, and two collections - the Wall Street Journal (1987-92, 173,255 articles, 21 MB) for English, and Nikkei Shinbun (1991, 151,650 articles, 178 MB) for Japanese. The

strategies are organized in the way of i) the choice of fields, ii) the weighting scheme, and iii) the synonym handling for <Con> keywords. Table 4 shows the result of this experiment. Although this is a data used an only single topic, several interesting phenomena were observed:

(1) The correlation among strategies was very weak (=0.16) in contrast to a strong correlation (=0.99) among the effects of phrasal operators for short queries. The best Japanese query strategy could be quite different from the English one; (2) Japanese strategies showed more variability in effectiveness. This Japanese result shows a contrast to the data of phrasal operations for short queries in Section 3.4 where any of them didn't work well consistently; (3) Adding fields (+<Desc> and +<Con+Fac>) doesn't show significant improvement in both languages. (Adding <Narr> harmed the performance in both languages [data omitted]); (4) The linear weighting is reasonably effective in the two languages; (5) The query expansion by the synonym operator was effective in Japanese, but not in English. Based on this and the thesaurus effect of the character-based indexing (section 3.6), Japanese could possibly be called a *thesaurus effective language*; (6) The "English Method", which had been empirically crafted, was most effective in English.

Table 4. Effectiveness of Various Strategies in Japanese and English (Query="German Joint Ventures", Top 100 precision)

	Japanese (%inc)		English (%inc)		
#1)	31	(0)	48	(0)	<title> [=Baseline]
#2)	51	(+65)	44	(-8)	<title> with #Syn
#3)	22	(-29)	38	(-21)	Unique<title+Desc>
#4)	30	(-3)	52	(+8)	Linear<title+Desc>
#5)	35	(+13)	48	(0)	Linear<title+Desc+Con+Fac>
#6)	34	(+10)	50	(+4)	Square<title+Desc+Con+Fac>
#7)	43	(+39)	50	(+4)	Square<title+Desc+Con+Fac> with #Syn
#8)	37	(+19)	36	(-25)	Square<title+Desc+Con+Fac> with #Max
#9)	38	(+23)	57	(+19)	English Method [= Double<title>+ <Desc+Con>+Double<title>+<Desc+Con+Fac>+ #Uw50<title>+(#Uw50<title> with #Syn)]

Correlation Coefficient = 0.157

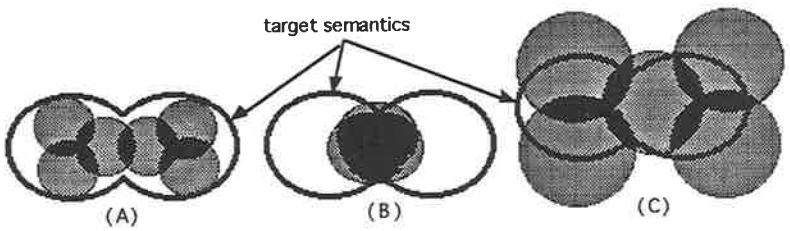


Figure 5. Three Kinds of Semantic Coverage.
((A) improving, (B) no change, and (C) getting worse)

As in Figure 5, the above retrieval behaviors can be conceptualized in terms of individual semantic specificity (size of each circle) in the context, and the total coverage of semantics (distribution of circles). When query semantics is narrowly specified by lexical entries, or it is specified locally by phrases, as we saw in Japanese before, the coverage of the target semantics by the query expansion will be thesaurus effective.

5. Summary

Briefly summarizing the above discussions: 1) Although the inference network works well for both English and Japanese, Japanese performs better than English for short queries because of its lexical semantic specificity; 2) Word distance has less effect in Japanese because of its locality. Classifying Japanese lexical and syntactical words may be effective to control the locality problem; 3) Mixed-mode indexing takes the advantages of the character-based and word-based; 4) Good strategies for a Japanese long (e.g., TIPSTER-type) query will be very different from English. Japanese, as a thesaurus effective language, performed well with synonym expansion, and the "English Method" worked best in English, but not in Japanese.

Acknowledgment

We wish to express our thanks to Chisato Kitagawa for his supportive discussions of this paper. This research was supported by the NSF Center for Intelligent Information Retrieval at the University of Massachusetts at Amherst.

References

Callan, J. P., Croft, W. B., Harding, S. M., "The INQUERY Retrieval System", 3rd International Conference on Database and Expert Systems Application, pp. 78-8, 1992.

Callan, J. P., Croft, W. B., "An Evaluation of Query Processing Strategies Using the TIPSTER Collection", ACM SIGIR-93, pp. 347-355, 1993.

Croft, W. B., Turtle, H. R., Lewis, D. D., "The Use of Phrases and Structured Queries in Information Retrieval", ACM SIGIR-91, pp. 32-45, 1991.

Fagan, J., "Experiments in Automatic Phrase Indexing for Document Retrieval - A Comparison of Syntactic and Non-Syntactic Methods.", Ph.D. Dissertation, Cornell University, 1987.

Fujii, H., Croft, W. B., "A Comparison of Indexing Techniques for Japanese Text Retrieval", ACM SIGIR-93, pp. 237-246, 1993.

Harman, D., "The DARPA TIPSTER Project", SIGIR Forum, 26(2), pp. 26-28, 1992.

Kageyama, T., "The Place of Morphology in the Grammar: Verb-Verb Compounds in Japanese", in Yearbook of Morphology, (eds.) Booij & van Marle, 1989.

Kajiwara, K., "History of Words for the Thermometer in Japanese: Changes and Acceptance of Modern Chinese Words (A type)", The National Language Research Institute Research Report, 105(14), pp. 81-137, 1993.

Kimoto, H., Tanaka, T., Ishikawa, T., et al., , A Proposal for Constructing a Test Collection for Information Retrieval Systems, IPSJ JohoGaku Kiso 32(1), pp. 1-8, 1993.

Matsumoto, Y., Kurohashi, S., Myoki, Y., et al., "User's Guide for the JUMAN system - A User-Extensible Morphological Analyzer for Japanese", Nagao Lab., Kyoto University, 1991.

Matsuo, J., Nishio, T., Tanaka A., " Japanese Synonymy and its Problems", The National Language Research Institute Report 28, Shuei-Shuppan, Tokyo, 1965.

Turtle, H. R., "Inference Network for Document Retrieval", Doctoral Dissertation, University of Massachusetts, 1991.

Turtle, H. R., Croft, W. B., "Evaluation of an Inference Network-based Retrieval Model", ACM Transactions on Information Systems, 9(3), pp. 187-222, 1991.

Appendix 1. A Procedure to Create Test Collections for English-Japanese Comparative Experiments

I) From some selected texts in both languages which contains the same information, measure the sentence length and the ratio. For this task, we used the English and Japanese pamphlets of Smithsonian museums [title: *Smithsonian Institute*, 1991 revision]. The result was 1 : 2.5 in characters for Japanese vs. English, i.e., 1 : 1.25 in byte length.

II) Choose a collection of one language, and get the statistics of text length frequencies. We used a Japanese collection of business newspaper articles about "joint ventures", which contains 890 documents.

III) Create a English population of documents of the same subject. We made this from a *Wall Street Journal* database of the corresponding years to the Japanese documents [year: 1987-91, size: 498 MB, 163,092 documents], giving an INQUERY query, "joint venture".

IV) Using the text length frequencies of Japanese as a probability distribution, choose a set of English articles randomly from the population.

Appendix 2. A Sample TIPSTER Query

<top>
<head> Tipster Topic Description
<num> Number: j01mod
<dom> Domain: 国際経済 [International Economics]
<title> Topic: ドイツの合弁 [German Joint Ventures]
<desc> Description:
文書ではドイツ企業による新合弁について報告する。
[Document will announce a new joint venture involving a German company.]
<narr> Narrative:
該当文書ではドイツの会社と ..... [A relevant document will announce a new joint venture involving a German company. Any form of the venture is acceptable. For example, a joint establishment of a new company, or a joint development of a new product, etc. But, the document must identify the names of German companies, and the name of the product or the service.]
<con> Concepts:
1. 合弁, 提携, 共同, 連携, 協力
[joint venture, tie up, partnership, cooperation, collaboration]
2. 会社, 企業, 事業 [company, enterprise, business]
3. 合弁会社 [joint concern]
4. ドイツ, 独 [Germany, German, Deutsche]
<fac> Factor(s):
<nat> Nationality: ドイツ [Germany]
</fac>
</top>

A PHRASE-RETRIEVAL SYSTEM BASED ON RECURRENCE

Magnus Merkel, Bernt Nilsson & Lars Ahrenberg

Department of Computer & Information Science
Linköping University
S-581 83 Linköping
SWEDEN
email: {mme,bkn,lah}@ida.liu.se

Abstract

The paper describes a simple but useful phrase-retrieval system that primarily is intended as a support tool for computer-aided translation. Given no other input than a text (and a word list used for filtering purposes), the system retrieves recurrent sentences and phrases of the text and their positions. In addition the system provides information on internal and external recurrence rates.

1. Introduction

As any localiser knows, manuals and other types of documentation are often highly repetitive, i.e. many phrases, in fact even sentences and sometimes paragraphs, tend to recur in the text. This is a fact that can obviously be used to speed up translation. By combining two sets of data for a recurrent segment, viz. a record of the positions in the text where it is found, and a record of its translation (assuming it to be unique), all occurrences of the segment can be translated in one go. As the recurrence rate within one document (*internal recurrence*) can sometimes be higher than 25% (see Table 1), and even higher if the document is compared, say, with a family of documents or a previous version of the same product (*external recurrence*), the gains in speed, costs and consistency can be quite substantial (see Table 2 and 3). In fact, it seems that translation aids of this kind are already in use by some translation companies, see e.g. Language Industry Monitor (1994). Moreover, information on recurrent segments and their frequencies is useful in combination with any type of memory-based translation system, as it tells you what would be useful to have in the system's database(s).

To make the scheme work, however, you need a system that finds the recurrent segments of a given document for you. Moreover, to reduce the time required for manual inspection and correction of the generated list of segments, the system should only generate segments that are appropriate translation units. It is such a system that this paper is about.

The system, nick-named FRASSE, actually has two primary uses. On the one hand it can be used as a data acquisition tool for a memory-based translation system, as described above.

It then actually derives only half of the needed data, as the translations have to be acquired in some other way. On the other hand it can be used diagnostically to generate a profile for a given document, calculating internal and external recurrence rates and frequencies for recurrent segments. This profile can be taken as a basis for determining whether a translation memory should be used at all, and if so, with what (initial) content in its databases.

Apart from its use in computer-aided translation FRASSE could also be a useful tool for a lexicographer or a literary student. Most systems with a functionality similar to FRASSE's, such as Choueka (1988) and Lessard & Hamm (1991) have apparently been developed for such purposes. The contributions of FRASSE are that the search process is exhaustive given a text as input and that phrase retrieval can be done on untagged texts. Furthermore, there is a measuring function which calculates how much of a given text is made up from recurrent segments and sentences. Large texts can be split up into smaller subtexts and analysed separately for reasons of complexity, but the retrieved segments from each subtext can be compared with other data to find differences or to combine subtext data into data for the whole text. FRASSE has been used on texts of 1 million words and can be run on both UNIX systems and PCs.

Internal Recurrence	**SS1**	**SS2**
Total no of words	254,350	248,089
Sentence Recurrence	25 % (2,290 sents)	15 % (1,822 sents)
Segment Recurrence	29 % (1,824 phrases)	31% (1,911 phrases)
Sentence & Phrase Recurrence	43 %	39 %

Table 1. Internal recurrence for User's Guides of two versions of the same spreadsheet program, SS1 and SS2 (from Merkel 1992). The phrase segments have been revised manually and segments which do not function as translation units have been removed.

External Recurrence in SS2 relative to SS1	
Shared sentences	3,677
External Sentence Recurrence	20 %
Shared Segments	620
External Segment Recurrence	15 %
External Sentence + Segment Recurrence	31 %

Table 2. External recurrence in SS2 relative to SS1

Combination Internal/External Recurrence	**SS2**
Internal + External Sentences	33 %
Internal Sentences + External Sentences + External Segments	42 %
Internal Sentences + External Sentences + Internal Segments	52 %

Table 3. Combination of internal and external recurrence in SS2 (relative to SS1)

The rest of this paper is organised as follows. Section 2 gives a short presentation of our goals for FRASSE within the framework of a project on computer-aided translation that we

are currently engaged in. We also define some of the notions we will use in the sequel. Section 3 describes the functionality of the system and section 4 illustrates how it can be used. In section 5 we compare FRASSE with some other similar systems and in section 6, finally, we discuss our plans to improve it.

2. On the use of recurrent segments in translation

A basic assumption underlying our translation scheme is that a recurrent segment in the normal case is a translation unit. This assumption does not always hold, however. If recurrence is the only criterion, we are likely to identify segments that should really be treated differently, because they have a different analysis in different contexts. The segment "the file" should be treated differently if it is used in the context of (1) "the file manager", (2) "the file menu" or (3) "the file" when it is translated into Swedish. In context (1) it would be translated as "Fil(hanteraren)", in (2) as "Arkiv(-menyn)" and in (3) as "filen". Also, as will be shown below, we will find a good deal of linguistic nonsense that needs to be filtered out. It is thus important to design the system so that only relevant recurrent segments are found, which means that we have to find some criteria that distinguish the relevant segments from the irrelevant ones. Related to this question is the level of abstractness of the segments; are they to be represented as unanalysed word segments, lemmatised word segments or perhaps something more general such as phrase patterns or phrase structure rules?

Another important question is whether consistency in translation of recurrent segments is always to be preferred, as it is for terminology, or whether there are situations when different translations should be used, even though the source segment is a recurrent unit of the source text. In a small pilot study on aligned parallel texts (Larsson & Merkel, 1994) we found a few cases where translators had chosen different translations for the same segment of the source text, but with no apparent harm.

English: *Select the port you want to use.*
Swedish 1: *Välj den port du vill använda.*
Swedish 2: *Markera den port du vill använda.*

Apart from the dimensions of consistency and variation, one may suspect that the use of fixed translations for recurrent segments, may affect the coherence of the target text. This is a question that we intend to investigate in our further work.

We end this section with some definitions of the terms that we use in the paper.

- A *segment* is a sequence of two or more consecutive words that does not cross sentence boundaries.
- A *recurrent segment* is a segment that occurs in at least two different sentences of a given text or corpus. For practical reasons, we often use a higher lower bound

than two for the number of occurrences, but there is no non-arbitrary way to fix this lower bound. Note that a recurrent segment as such may be a proper syntactic unit, or a collocation, but that it can be neither.

- A *phrase* is a segment constituting a syntactic unit with compositional meaning.
- A *phrase pattern* is a sequence of words and categories that a phrase can be analysed as (e.g. *Select from *).
- A *collocation* is a segment constituting a lexical unit of some sort.
- A *translation unit* is a simple or complex linguistic unit of a source language that can be associated with a set of corresponding units of a target language. Different instances of a translation unit in a text may have different translations, but it must be possible to assign it some corresponding instance in the target text.

3. The FRASSE system

The program is implemented in standard C and there are currently compiled versions for UNIX, Windows NT and OS/2. The following functionality is included in FRASSE:

3.1 Retrieval of recurrent sentences

Sentences are identified in a technical sense, i.e. they are identified by punctuation information and carriage return characters.

Input: Text

Output: A segment table with sentence types (as word strings), position (document identifier and a list of offsets) and number of occurrences for each sentence in the text.

3.2 Retrieval of recurrent segments

Recurrent segments are retrieved from the text after it has been split up into sentence types. Either you can input the text (Frasse will then create a table with sentence types, see 3.1) or one or more existing segment tables. The data structure for storing segments consists of a table of segments. Each entry in the table contains the following fields:

1. *The text segment*
2. *A list of locations* where each location contains a document identifier and a list of offsets inside each document.
3. *A parsed word string*, a sequence of references to the word table, see below. All text segments are first parsed to word strings so that comparisons of words can be done by just comparing pointers. (This field is not present in the output.)

In addition to the above fields, the data structure of the segment table holds a *word table*, i.e. a table of known words with frequencies.

Each recurrent segment is stored in only one entry, but has multiple locations with multiple offsets.

Input: Text or a list of segment tables
Output: A new segment table holding maximal segments from input.

All combinations of segments from the input table are compared pairwise. Each pair, *<s1,s2>*, is searched for common parts. A common segment is defined as all segments that contain the same consecutive words in both *s1* and *s2*. If the identified segment is long enough it is stored with all locations and adjusted offsets from both *s1* and *s2*.

We only consider segments that start at locations such that the immediately preceding words, if any, are different in *s1* and *s2*. A segment ends in the same manner. This means that only the longest possible segments are considered. Note that both *a b c d* and *a b c d e* can be regarded as maximal segments if they have different frequencies, i.e. if *a b c d* occurs 15 times and *a b c d e* 12 times they are both maximal segments.

The fact that only the longest possible segments for each pair are stored has the effect that there is a significant reduction of data, which makes the algorithm more efficient.

There is a filter that trims the segments at the head and tail of the segment, which reduces the size of the segment table. The filtering strategies are discussed further in section 4.

The resulting segment tables can be sorted in various ways (any combination of sorting by alphabetical order, length of segments and frequency). It is also possible to strip away the location information, which leaves a list of segments and their corresponding frequency data. The stripped segment list is easier to handle when a translator is viewing or revising the segments. After revision it is possible to extract the location information for each segment in the revised segment list and create a new complete segment table.

3.3 Measuring the recurrence rate

Given a segment table FRASSE can return the recurrence rate for the text.

Input: A segment table with recorded positions
Output: (1) Total number of words in the analysed text; (2) number of words that were part of recurrent segments, and (3) The ratio of (2) to (1) expressed as a percentage.

The algorithm calculates the recurrence rate by positioning all segment types on top of the actual text. The fact that segments may be overlapping is represented in the resulting figures.

3.4 Comparison and combination of several segment tables

It is possible to compare and combine several sets of analysis results (i.e. the intersection or union of different analysis results). This could be useful for large text materials which could be split into subtexts, but where it would be useful to compare the data.

Input: Segment tables, *st1, st2... stn.*
Output: A new segment table which holds the intersection or union of segments in the input tables.

3.5 Performance

The program has been run on texts of various sizes, up to 1 million words. Sentence retrieval is very fast (under a second up to a minute depending on the size of the text), but segment retrieval is considerably slower.

The segment retrieval algorithm has a quadratic time complexity in terms of the number of sentence types and a worst case quadratic complexity in terms of the maximum sentence length. If *m* is the number of sentence types (or segments) in the initial table, the program has to consider *m(m-1)/2* pairs. For a given pair of sentence types with sentences of length *n* and *p*, respectively, the program considers *np* positions as potential starting posititions for a common segment. This happens regardless of the required minimal length of a recurrent segment, since, fo a normal text, only a fraction of the potential starting posititions will require a large amount of processing.

In a test run we doubled the text size and the initial table twice. There were no common segments in the first text and the texts that were subsequently added to it. The test was made on a PC running Windows NT. In the first run a text comprising 25,000 words and 1,765 sentence types was used, which took 45 seconds (for segments of length 2 and larger). In the second run the size was doubled and this time the program needed 3 minutes 15 seconds. The final run made on 100,000 words and 7,060 sentence types took 12 minutes 15 seconds. This agrees with our previous experience of the program and a rough estimate is that 200,000 words of repetitive text takes about 50 minutes to process, 400,000 words 3 hours 15 minutes, etc., given that the program has enough virtual memory available.

When we analysed a legal text of 830,000 words the results were that 32.73 per cent of text were made up by recurrent sentences. The segment analysis yielded that the overall coverage of unrevised recurrent segments made up 73.66 per cent of the full text. The analysis was of course relatively time and processing intensive, so we also divided the text into three subsections which in some sense were related in content and, as expected, the result was that the coverage ratio decreased, but surprisingly little. The coverage ratio varied from 65.81 per cent up to 69.69 per cent for each subtext. The small difference between a full analysis and several partial analyses indicates that a large text does not necessarily have

to be analysed in one batch, especially since we have the possibility to compare the results from the partial analyses to find common data.

4. Example from using FRASSE

An example of the segments (length 3 or longer) retrieved by FRASSE is shown below. Note that this list is retrieved without the use of any filters and that it therefore reflects the maximal strings in the source text in a very crude way. The segments could not be fed into a translation memory as they are here; a manual revision or complementing filtering would be necessary.

you want to	452	and then choose	82
, you can	392	the database window	80
for example,	327	in this chapter.	79
menu, choose	136	, click the	79
if you want	130	the edit menu	78
you can use	119	you can also	78
to create a	112	the tool bar	77
, and then	109	a form or	73
example, you	106	in design view	73
if you want to	105	choose the ok button	72
, see chapter	105	you can create	72
for example, you	102	the ok button.	71
in this chapter	100	, select the	71
the qbe grid	99	then choose the	70
form or report	94	choose the ok button.	69
, see "	88	for more information	69

Table 4. The top 32 segments generated by FRASSE from a computer program User's Guide (without filtering)

A problem that arises is what criteria should be used when deciding translation units. When we revised the output from FRASSE by hand we found that a majority of the segments that we removed from the original output actually were segments that ended with function words (such as "and", "but", "to", "in", etc.) and segments with only one parenthesis character or quotation mark. Therefore we implemented a filter where it is possible to define words that should be stripped at the beginning and at the end of segments as well as requirements on what kinds of characters that should be regarded as pairs (quotation marks, parentheses, etc). In table 5 below, the result is shown when we applied such a filtering mechanism on the same text as in table 4. We used function words and frequent verbs which have a very general meaning in the text type. Of the 32 most frequent segments in the unfiltered result above, 22 have been filtered out, leaving a residue of 10.

in this chapter	100
the qbe grid	99
form or report	94
the database window	80
the edit menu	78
the tool bar	77
the ok button	74
in design view	73
choose the ok button	72
for more information	69

Table 5. The top 10 segments generated by FRASSE (with filtering)

The strategy has a preference to keep noun phrases and prepositional phrases as well as longer segments ending in a non-function word.

4.1 Applications of FRASSE

As we have pointed out earlier the intention of FRASSE is to enhance working with translation memory-based translation tools. The program can also be seen as a diagnostic tool where it is possible for a translation co-ordinator to decide what recurrence profile a given text has. You could compare the document to be translated internally (i.e. by calculating the recurrence rate within the document) or externally (i.e. by comparing the document with related documents that have already been translated and which are stored in existing translation memories). A concrete application for FRASSE would be to construct terminology lists consisting of frequent segments that can be stored in the phrase lexicon of a translation memory tool. It is not necessary to translate them before the translation starts. Instead the translations of the recurrent phrases can easily be made the first time they occur in the source text, making the translation data available the next time the phrase is encountered.

Another application would be in lexicography which is the main aim for systems like XTRACT, mentioned below. One object for a lexicographer is to identify collocations in a text that could be considered as entries in a domain-specific dictionary. As collocations are arbitrary, dependent on the text domain and recurrent, they can in principle only be recovered from corpus material and by performing statistical and linguistic analysis of their occurrence (see Smadja, 1993 for a more elaborate discussion).

Researchers involved in literary analysis would also find analysis of recurrent segments useful when they try to find characteristics of a work of literature. Lessard & Hamm (1991) describes a program that was designed for these purposes. However, they report some shortcomings with their program, e.g. subsegments are not always removed when maximal strings have been identified. Furthermore, their program cannot handle a text as the sole input. First a concordance has to be produced for each lexical item in the text and then this concordance is analysed for repetitive structures. This also meant that they couldn't analyse large texts in a single pass, because the concordance information wouldn't fit into memory.

When companies or institutions produce bulky documentation which requires a team of writers working on the same set of documents, some level of standardisation must be achieved. By analysing already existing material with tools like FRASSE, it would be possible to standardise terminology and phraseology. A segment table from FRASSE sorted alphabetically is an excellent starting point in pin-pointing inconsistent phraseology.

5. Comparisons to related work

Similar systems have been described in other fields of NLP, especially within lexicography and literary analysis (see previous section). We would like to mention the following research and point out the differences with our approach.

5.1 Choueka's n-gram analyser

In Choueka (1988) an n-gram analyser is described which produces collocational expressions of 2 to 6 words. The corpus that was analysed consisted of ten million words of newspaper text. Only sequences of words of length 2 to 6 with a frequency above 10 were retrieved. After the initial analysis the raw list of collocations was filtered in different stages to reduce the size of it. There are two main restrictions in Choueka's work compared to our approach, namely that the maximal segments are not singled out from subsegments and that there is an upper limit on the length of collocational expressions.

5.2 Smadja's XTRACT

XTRACT developed at Columbia university extracts collocations from a parts-of-speech tagged text (Smadja 1993). The program works in several steps, building significant bigrams and expanding these to n-gram collocations. The program retrieves the longest significant collocations, but in order to take full advantage of the algorithm the text that is being analysed must be tagged. In the current version of XTRACT it is not possible to use a text as the only input to the program. You must specify each *seed word* that you want to test for significant bigrams. There are several interesting ideas in Smadja's approach, especially the overall intent to filter out non-significant combinations of words as early as possible in the analysis process in order to increase the efficiency and relevance of the output data.

Apart from Choueka and Smadja, work on identifying collocations in large texts have been made by for example Church & Hanks (1989).

6. Future work

As pointed out earlier in this paper there are several aspects of our approach which we would like to develop further. Here are some examples of what we are working with:

- *Improving the efficiency* of FRASSE. We are confident that the performance of the program could be improved considerably by, for example, using word frequency information to block irrelevant segment processing.
- *Creating a working environment for reviewing translation units.* This entails the possibility to select a segment from a segment table and look at its different contexts in the source text, i.e. a kind of concordance facility.
- *Combining FRASSE with an alignment system* to make it possible to make systematic recurrence analyses of parallel texts.
- *Validating translation units* by running FRASSE on a parallel English-Swedish corpus that we are currently setting up.
- *Retrieval of 'abstract' units* such as phrase patterns, lemmatised segments and conceptual units, e.g. the segment "Press the ESC key" could be analysed as

'Press NP' or "Press <key>". One idea that we would explore further in this context is to use *seed segments* (as a complement to *seed words*) to find relationships between different segments within sentences.

FRASSE is available to the research community via Internet at the time of publication of this paper. Information on how to download the software can be obtained from the authors. The availability of FRASSE will be announced on appropriate discussion lists.

References

Choueka, Y. (1988) "Looking for needles in a haystack". In *RIAO 88, User-oriented Content-based Text and Image Handling*, Volume 1, 609-623, 1988.

Church K. & Hanks, P. (1989). "Word association norms, mutual information and lexicography." In *Proceedings from the 27th Meeting of the ACL*, 76-83.

Language Industry Monitor No. 19 (1994). "Mendez Rolls its Own".

Larsson, A. & Merkel, M. (1994). "Semiotics at Work: Technical Translation and Communication in a Multilingual Corporate Environment". To be published in the *Proceedings of NODALIDA* (Nordiska Datalingvistikdagarna), Stockholm university.

Lessard, G. & Hamm, J.-J. (1991). Computer-Aided Analysis of Repeated Structures: the Case of Stendahl's Armance. In Journal of the Association for Literary and Linguistic Computing., Vol. 6, No. 4, 1991.

Merkel, M. (1992). Recurrent Patterns in Technical Documentation. Research Report LiTH-IDA-R-92-31, Dept. of Computer and Information Science, Linköping University.

Merkel, M. (1993). "When and why should translations be reused?" In *Papers from the XIII VAAKKI symposium 1993*, Vaasa.

Smadja F. (1993) "Retrieving Collocations from Text: Xtract". In *Computational Linguistics*, vol. 19, no. 1.

Automatic Sublanguage Identification for a New Text

Satoshi SEKINE
Computer Science Department
New York University
715 Broadway, Room.709
New York, NY 10003, USA

Abstract

A number of theoretical studies have been devoted to the notion of sublanguage, which mainly concerns linguistic phenomena restricted by the domain or context. Furthermore, there are some successful NLP systems which have explicitly or implicitly addressed the sublanguage restrictions (e.g. TAUM-METEO, ATR). This suggests the following two objectives for future NLP research: 1) automatic linguistic knowledge acquisition for sublanguage, and 2) automatic definition of sublanguage and identification of it for a new text. The two issues become realistic owing to the appearance of large corpora. Despite of the recent bloom of the research on the first objective, there are few on the second objective. If this objective is achieved, NLP systems will be able to optimize to the sublanguage before processing the text, and this will be a significant help in automatic processing. A preliminary experiment aiming at the second objective is addressed in this paper. It is conducted on about 3 MB of Wall Street Journal corpus. We made up article clusters (sublanguages) based on word appearance, and the closest article cluster among the set of clusters is chosen for each test article. The comparison between the new articles and the clusters shows the success of the sublanguage identification and also the promising ability of the method. Also the result of an experiment using the first two sentences in the articles indicates the feasibility of applying this method to speech recognition or other systems which can't access the whole article prior to the processing.

1 Introduction

A number of theoretical studies have been devoted to the notion of sublanguage, which mainly concerns linguistic phenomena, including syntax, semantics or pragmatics, restricted by the domain, context or

discourse of the text or the utterance. In particular, sublanguage studies indicated that several kinds of restrictions or deviations are characteristic for each sublanguage [1] [2] [3] [4].

Furthermore, there are some successful natural language processing systems which have explicitly or implicitly utilized sublanguage restrictions. For example, TAUM-METEO [5] is a machine translation system in which the translation is aimed only at sentences in the weather forecast domain, and it works remarkably well . Also, recently ATR [6] built a translation system for the conference registration task, and it works well, too.

This suggests the following two objectives in order to make a breakthrough on current NLP research.

1. Automatic linguistic knowledge acquisition for sublanguages.

2. Automatic definition of sublanguages and identification of the sublanguage of a new text.

These two goals become realistic owing to the appearance of large corpora. Using large corpora, preliminary experiments to meet the first objective have been conducted [7] [8]. Although these are still small experiments, in terms of the accuracy and the coverage for practical applications, their objectives address the first goal above, and could make a breakthrough for future N.L.P. systems by reducing the costly and errorsome linguistic knowledge encoding task by human linguists.

The second objective has not received so much attention. In the previous sublanguage N.L.P. systems, the domain the system is dealing with is predefined. For example, the definitions of "weather forecast domain", "medical report domain" or "computer manual text" are artificially or intuitively defined by a human. This is actually one method to define the sublanguage of the text, and it seems to work well. However, it is not easy and not always possible. The processing of newspaper articles is one such example. Since the range of articles is normally wide, a human can't prepare or intervene at each article to decide which sublanguage the text belongs to and it is impossible to utilize the sublanguage knowledge. In this paper, we will propose a objective way of defining sublanguage and automatically identifying the sublanguage for a new text. Then, comparison of the identified sublanguage and the test text will be reported, which reveal the method is promising.

2 Sublanguage

There has been a significant amount of research on the notion of sublanguage. In the literatures, sublanguage was defined by the name of the domain, (e.g. 'computer manual domain', 'weather report') or type of the document, (e.g. 'fiction', 'exposition'). Although comparative studies indicated that there exist several distinctive features in each sublanguage, there is no guarantee that these

features are uniform over a sublanguage defined in this way. The computer manual domain may contains several sublanguages in it. Therefore we need an objective and linguistic measurement of sublanguage to define it.

In our experiment, automatic article clustering based on word appearance is used to generate a set of sublanguages. The method we used is based on a document clustering metric [9] [10]. Although word appearance is one of the main features of sublanguage, other kinds of phenomena, including syntactic and semantic features, have been reported as sublanguage features. It would be better to utilize these phenomena as well in defining sublanguage, but several problems, in particular ambiguity problems, make it difficult for us to do so initially. Furthermore, word appearance is an important factor in N.L.P. systems. For example, word ambiguity in speech recognition and optical character recognition are good examples. These systems often produce several candidates for a portion of utterance or character sequence, and we sometime find that some of the candidates are totally irrelevant to the topic of the speech or the document (see the following example. Extracted from UNIX manual: command *banner*). This kind of ambiguity can be eliminated based on the sublanguage method which is being proposed in this paper.

```
Display  a  string in large letters.
Display aster ring in large lettuce.
```

3 Experiments

As we mentioned before, the experiments can be divided into the following three phases.

1. Sublanguage Definition: cluster articles based on word appearance
2. Sublanguage Identification: find the closest cluster to a new text
3. Evaluation: compare the cluster and the text

In addition, we will report another experiment which uses only the first two sentences of each article to determine its sublanguage. This experiment addresses the possibility of dynamic language adaptation in text processing.

4 Article Clustering

The corpus for the experiment consists of 1106 articles, extracted from a week of the Wall Street Journal (3 MB including header information). The statistics for the corpus are as follows:

- Number of articles: 1106
- Number of words (as tokens): 421281
- Number of words (as types): 25622

Clusters are produced by a similarity measure calculated between every two articles in the corpus. The formula for the similarity measure is based on the combination of inverse document frequencies of words and normalization by the number of the words in a text [10]. The formal definition similarity measure between article A_i and A_j is the following:

$$Similarity(A_i, A_j) = \frac{1}{|A_i||A_j|} \sum_{\{w|w \in A_i, A_j\}} \frac{1}{|A^w|} \quad (1)$$

Here, $|A|$ is the number of distinct words in article A, $|A^w|$ is the number of articles which contain word w. In the following explanation, values of the similarity measure are multiplied by 1,000,000, to aid readability.

The Cut off number for $|A^w|$ is set to 50, i.e. the sum is over words which occur in 50 or less articles throughout the corpus. This condition is introduced in order to avoid frequent words which may have no role in this calculation. Also, the minimum overlap is set to 3, i.e. the similarity measure between two articles becomes 0, if they have only 2 or less words in common. This condition is useful to avoid over-generation of accidental clusters.

Then a matrix of the similarity measure is obtained, whose values range from 0.0 to 1181.7 (recall that this value is 1,000,000 times the original similarity value). Clusters are generated based on this data. A simple algorithm, hierarchical clustering with single linkage method, is adopted for cluster generation. We set a threshold, i.e. any two articles for which the similarity measure is greater than the threshold belong to a same cluster. The value of the threshold is set to 50.0 experimentally. According to the algorithm above, we generated 129 clusters which have more than one article. The average number of articles in a cluster is 4.30 and the maximum number of articles in a cluster is 31.

5 Cluster Identification

We chose 50 test articles from new Wall Street Journal articles which are not used in the clustering experiment described above. In this section, we will describe the method to identify the closest

cluster for each test article. It is basically the same method as the calculation of the similarity measure explained in the previous section. For each test article, similarity measures to all clusters are calculated. Here, the similarity between an article and a cluster is set to equal to the maximum similarity between the article and an article in the cluster. This calculation is not so expensive, because we have a limited set of words which have to be taken into account in the calculation. The number of the words is normally much less than the number of tokens in the test articles and the number of clusters to be examined for each word is less than the cut off number (50 in the experiment). So this calculation is almost linear in the number of articles if there is enough space to store the word index.

We also set a threshold to decide if the test article and the cluster are similar enough. The threshold is set at the same level as the threshold which is used to produce the clusters (50.0). The result of the cluster identification experiment is shown in Table 1.

	Number of articles
Found the closest cluster	21
Found the closest article	9
Can't find anything	20

Table 1: Results of cluster identification

In Table 1, "Found the closest cluster" means that the closest cluster which contains 2 or more articles is found with greater similarity measure than the threshold value (50.0). "Found the closest article" means that it found the closest cluster satisfying the threshold, but the cluster contains only one article. "Can't find cluster" means that the closest cluster can't be found because of the threshold.

6 Evaluation

Evaluation of the experiment will be described in this section. The 21 articles which find the closest cluster containing more than 2 articles are examined at this evaluation. The evaluation measure is based on how many tokens in the test article also exist in the closest cluster, i.e. tokens which overlap between the test article and the closest cluster. A straightforward measurement is its coverage, which counts how many tokens are overlapping out of all the tokens in the article. This figure could help to decide how useful this method is, but it might still difficult to make an objective decision. So the number of overlapping tokens is compared with an expected value computed on the condition that tokens are randomly distributed in the 3 MB corpus (keeping word frequency distribution are the same). This expected value is calculated by the following formula.

$$E = \sum_{w} \frac{n_t * n_w}{N} (1 - (1 - \frac{n_c}{N})^{n_w}) \quad (2)$$

Here, n_t is the number of tokens in the test article, n_c is the number of tokens in the cluster, N is the number of tokens in the entire corpus, and n_w is the frequency of word w. The first factor inside the sum shows the expected frequency of a word w in the test article. The second term shows the probability that the word occurs at least once in the cluster in which n_c tokens exist. So the sum of the product for all the words in the corpus computes the expected number of tokens occuring in the test article which also occur in the closest cluster.

It is well known that high frequency words like "the" or "of", occur constantly regardless of the topic. On the other hand, low frequency words, which are often regarded as reflecting the topic, are expected to concentrated in similar articles. So, we can anticipate in this experiment that many low frequency words in the test article can also be found in the closest clusters. (Note that, because words with $|A^w| < 50$ are used in the similarity calculation, those words must be specially treated in the evaluation.) To observe such details, we classified the result by word frequency obtained on the 3MB corpus. The expected occurrence can be classified by limiting words in the sum to those whose frequencies are in the given range. Table 2 shows an example of the result. In the sample result, the number of tokens in the article is 164 and the number of tokens in the cluster is 621.

The first column shows the number of tokens and its expected number in the bracket below each number. For example, the expected number of tokens whose frequency ranges from 100 to 299 is 26.7, but is actually 24 in the article. These two figures indicate the balance of the word frequency in the article, and the averages of the ratio over all the 21 test articles are from 0.90 to 1.14, so we can say the articles are well balanced.

The second column shows information about overlapping tokens. For example, 13 tokens out of 24 tokens in frequency range from 100 to 299 are overlapping, and 106 out of 164 in the article are overlapping to the closest cluster.

The third column shows the coverage of overlapping tokens in the test article. We can see the overall coverage was 64.6% in this sample.

The figure in the fourth column indicates that tokens in the article are overlapping 135% of the expected value. Also 999% and 678% of expected overlap tokens are found in the article in frequency ranges from 1 to 49 and from 50 to 99, respectively. As mentioned before, the result for words whose frequency ranges between 1 to 49 can not be evaluated directly by the figure. As the method to find the closest cluster is based on similarity to each article, so the closest article in the cluster to the test article is the key in the cluster search. The number of tokens overlapping between the test article and the closest article in frequency 1 to 50, i.e. number of tokens used in the similarity calculation, is 6

Word frequency	Number of words	Overlap words	Coverage (%)	Ratio (%)
0	10 (5.6)			
1-49	32 (42.5)	11 (1.1)	34.4	998.8
50-99	22 (14.6)	10 (1.5)	45.5	678.2
100-299	24 (26.7)	13 (6.2)	54.2	209.0
300-999	22 (19.3)	18 (10.8)	81.8	167.2
1000-	54 (60.9)	54 (58.9)	100.0	91.6
Total	164 (164.0)	106 (78.5)	64.6	135.0

Table 2: An example of the result

for this sample (not shown in the table). This means that out of 11 overlapping tokens between the test article and the closest cluster, 6 tokens are found in the closest article and 5 others are found in the rest of the articles in the cluster. So these figures show the benefit of clustering in increasing the overlapping tokens.

The average coverage and ratio between the number of the overlapping tokens and the expected value throughout the 21 sample articles are given in Table 3.

The second column of the table shows the average coverage over the 21 articles in the experiment, and the third column shows the average ratio of number of overlap tokens to expected overlap tokens. From the figures in Table 3, we can tell the success of the method. For example, tokens whose frequency range from 50 to 99 are found with 473% of the expected value in the closest cluster. As these words are not used in the similarity calculation, it proves the existence of sublanguage at least with respect to word distribution. The same thing can apply to the data in range from 100 to 299.

For the words whose frequency ranges from 1 to 49, the total number of overlapping tokens between the test articles and their closest clusters is 256, and the number of overlapping tokens between the test articles and their closest articles, i.e. tokens used in the similarity calculation is 198. So, 58 new

Word Frequency	Coverage (%)	Ratio (%)
1 - 49	36.29	1077.56
50 - 99	46.25	473.45
100 - 299	54.11	230.16
300 - 999	78.52	140.47
1000 -	95.36	108.33
Total	67.18	142.04

Table 3: Average coverage and ratio

tokens are introduced by the clustering effect. These may be useful in language processing based on this sublanguage method.

Examining the values of the ratio in Table 3, it is intuitively understandable that the lower frequency words tend to have a large ratio and the higher frequency words tend to be close to 1.0. For instance, almost all of the words whose frequency is more than 1000 are closed class word, like "the", "of" or "it" (there are only 46 words which have frequency over 1000). The result that the ratio is close to 1.0 indicates that the distribution of these words is balanced, as usually assumed. On the other hand, the ratio in lower frequencies shows that lower frequency words tend to occur together in similar articles. This is the very fact that we had hypothesized before the experiment.

7 Experiment with first two sentences

For some N.L.P. applications, it may be impossible to see all the sentences in an article before processing. For example, spontaneous speech recognition systems have to process each utterance at a time. We will call these applications 'dynamic applications' in comparison to 'static applications' in which the system can pre-scan the material before the actual processing. The algorithm described above can't apply to dynamic applications directly. Therefore we conducted an experiment using the first two sentences in the test articles, instead of all the sentences in the article, for finding the closest cluster. If it can find a good cluster (i.e. sublanguage) for the rest of the article by using only the first two sentences, dynamic applications can adopt this method for the processing of sentences after the second sentence.

For this experiment, since the number of sentences to be examined is reduced, the minimum requirement of overlapping words is set to 2 instead of 3, and the threshold of similarity in this

experiment is deleted. Table 4 shows the evaluation of this experiment. The results are derived from the 21 test articles which are also used in the previous experiment.

Word Frequency	Coverage (%)	Ratio (%)
1 - 49	31.68	892.14
50 - 99	44.18	453.12
100 - 299	50.20	218.00
300 - 999	74.55	120.84
1000 -	96.48	104.07
Total	65.46	130.81

Table 4: Average coverage and ratio

Surprisingly, the result is almost the same as the previous one. Actually, more than half of them select the same closest cluster as that selected in the previous experiment by using all the sentences in an article. This fact is also intuitively understandable, because the first sentences in an article normally indicate its topic. So the first two sentences could help to find its sublanguage. This result is strongly encouraging that this method can be applicable to speech recognition systems or others in which it is impossible to pre-scan all the sentences before processing.

8 Discussion

The main discussion will be on the notion of sublanguage. In comparison to the traditional definition of sublanguage, we propose an objective and empirical definition of sublanguage. As we discussed, statistical methods using a large corpora are surely useful to define a sublanguage. However, we are not against the traditional way of sublanguage definition, since it well appeal to our intuition and actually it is useful for a certain purpose. Rather, our contention is that the definition may be depending on the purpose of processing.

The Wall Street Journal, which we used in the experiment, is normally regarded as a homogeneous material and a sublanguage itself. However, the results of the experiment shows that we can find several clusters in it and an NLP system will benefit if the system regards WSJ as a bunch of sublanguages.

Since the algorithm has worked successfully on this homogeneous corpus, it may be interesting to apply it to more general materials. For example, a daily newspaper includes wider range of topics, so it is easy to imagine that this algorithm works better in such a corpus.

The clustering method is problematic. In the experiment, parameters are set to make clusters for which the average number of articles in a cluster becomes 4.3. The bigger the clusters are, the bigger the coverage might be, which means the more tokens overlap between a test article and the closest cluster. This leads sublanguage knowledge more useful, but at the same time amount of the knowledge become larger and more ambiguity may be contained. This trade-off has to be settled by the feature of sublanguage and also the feature of the system. This is not easy to solve, but has to be considered at its implementation.

9 Conclusion

In conclusion, it appears that the sublanguage definition and identification works by using large scale corpus, and the evaluation results show that the automatically found sublanguage knowledge is useful in processing of a new text. The second experiment prove that the method is applicable not only for static applications which can pre-scan the material before its actual processing, but also for dynamic applications which process sequentially the sentences from the top to the bottom, like a speech recognition system.

We can find several directions of future study. One is, of course, to refine the algorithm and to make larger experiment. Also an experiment with different kind of corpus could give us fruitful prospects. With regard to applications, as has been repeatedly mentioned, speech recognition is one of the most interesting. Although the utility of this approach may heavily depend on the characteristics of the speech recognition system to find out how useful this method is, we believe that a certain type of ambiguity can be resolved and furthermore it may become possible to enlarge the vocabulary size of the word dictionary. Finally, it might be interesting to explore the same technique towards not only word occurrence but also other kinds of linguistic knowledge, including syntax, semantics or others.

10 Acknowledgments

The work reported here was supported by the Advanced Research Projects Agency under contract DABT63-93-C-0058 from the Department of the U.S. Army. We would like to thank our colleagues at NYU, in particular Prof.Grishman, whose comments have been very useful.

References

[1] R.Kittredge, J.Lehrberger ed.: "Sublanguage: Study of language in restricted semantic domain" (1982)

[2] R.Grishman, R.Kittredge ed.: "Analyzing language in restricted domains" (1986)

[3] D.Biber: "Using Register-Diversified Corpora for General Language Studies" *Computational Linguistics Vol.19, Number 2* (1993)

[4] W.Gale, K.Church, D Yarowsky: "One Sense Per Discourse" *4th DARPA Speech and Natural Language Workshop* (1992)

[5] P.Isabelle: "Machine Translation at the TAUM group" *The ISSCO Tutorial on Machine Translation* (1984)

[6] T.Morimoto et al.: "ATR's speech translation system: ASURA" 42.2, *Eurospeech* (1993)

[7] R.Grishman: "Discovery Procedures for Sublanguage Selectional Patterns: Initial Experiments" *Comp. Linguistics Vol.12 No.3* (1986)

[8] S.Sekine, S.Ananiadou, J.J.Carroll, J.Tsujii: "Linguistic Knowledge Generator" *COLING-92* (1992)

[9] P.Willett: "Resent trends in hierarchic document clustering: a critical review" *Information Processing and Management Vol.24, No.5 pp.577-597* (1988)

[10] K.Sparck-Jones: "Index Term Weighting" *Information Storage and Retrieval, Vol.9, p619-633* (1973)

STRING COMPARISON BASED ON SUBSTRING EQUATIONS

Kyoji Umemura

Nippon Telegraph and Telephone, Basic Research Laboratories
Suite 4S222S, NTT, 3-1 Morinosato-Wakamiya, Atsugi, Kanagawa 243-01, Japan

Abstract

This paper describes a practical method to compute whether two strings are equivalent under certain equations. This method uses a procedure called Critical-Pair/Completion. that generates rewriting rules from equations. Unlike other Critical-Pair/Completion procedures, the procedure described here always stops for all equations because it treats strings of bounded length. This paper also explains the importance of the string equivalence problem if international data handling is required.

1. Introduction

The equality test is a fundamental operation. There are various kinds of equality functions. One function (i.e. STRING=) distinguishes lower case and upper case letter. Another function (i.e. STRING-EQUAL) ignores the case. Different kinds of equality test functions are necessary to handle various languages. For example, Japanese has Hiragana and Katanaka character sets. Although they may have different roles in written form, corresponding characters have identical pronunciations. This situation is very similar to lower and upper case letters in English. Therefore it is beneficial for Japanese people to have a function that ignores the Hiragana-Katakana difference. Futhermore, if the Japanese data is expressed in alphabet form, the equivalence is not simply between character and character, but between string and string. Other language may have same problem. This '*case-insensitive*' is not a simple problem for international data handling.

The case-insensitive comparison is the comparison under equations such as "A"="a", "B"="b", "C"="c", ... and "Z"="z". Since equations like "cha"="tya" is necessary in Japanese language, both sides of the equations should not be limited to one character. We can perform the case-insensitive comparison by replacing all upper case characters with the corresponding lowercase characters. In general, we compute equivalence by converting strings using some rules, replacing all of the substrings, obtaining canonical form and then comparing them exactly. These rules are usually generated by hand. It is a difficult task to generate these rules correctly for complex cases.

Suppose we have a equation set { "abc"="ab", "abc"="bc" }. The following strings are equivalent:"xabcabcy", "xababy", "xbcbcy", and "xabbcy". The rule set { "abc"->"ab", "abc"->"bc" }

is not correct since it is not confluent; For example "xabcy", may have two different results. The rule set {"ab" -> "abc", "bc" -> "abc"} is not usable since the conversion never stops. Though the rule {"abc"->"bc", "bc"->"ab"} looks adequate, the rule is not suitable for "xaaby" and "xbcy". The string "xaaby" is equivalent to "xbcy" because "xaaby" = "xaabcy" ="xabcy" = "xbcy". However, "xaaby" and "xbcy" are converted to "xaaby" and "xaby" using the rule set { "abc"->"ab", "bc"->"ab"}. The correct rule set that the procedure generates is {"aab"->"ab", "bc"->"ab"}.

In simple cases, like case-insensitive comparisons, the corresponding conversion rules are simple and apparent. However, the procedure that generates these rules from the equations are not simple. The procedure is one variation of the Critical-Pair / Completion procedure, which originally treats term-rewriting-systems. It is well known that Critical-Pair / Completion sometimes fails to stop. Even if we limit the domain to strings, the Critical-Pair / Completion procedure may not stop. For example, the equation {"aba"="ab"} will generate an infinite number of rules such as { "aba"->"ab", "abba" -> "abb", "abbba" -> "abbb" ... }. Since failure to stop is fatal for any application, we need a domain in which procedures always stops.

This paper formalizes the problem where strings are bounded in length. This problem is practical since the strings in computer system are bounded in length. This paper describes this variation of Critical-Pair/Completion procedure[Buchberger87]. This variation stops for all given equation. These characteristics are important for the actual application.

2. The Problem Definition

In this section, we define the problem to be solved.

N: N is a given number. N is the maximum length of any string.

C: C is a given finite set of characters. Any character has corresponding integer value.

String: String is a vector of the member of C.

N: SN is a set of all strings whose length is less than N. It includes the null string, which is the string whose length is 0.

G: G is a given set of equations among members of SN that defines the equivalence relation. An example of G is { "abc" = "bc", "abc" = "ab" }.

Substitution: Each of α, β, and γ is member of SN. Suppose we have a equation g in which $\alpha=\beta$ or $\beta=\alpha$, where α is a substring of γ. If there is a string δ that replaces the substring of γ with β and the length of δ is less than N, we call δ the substitution of γ with g.

Neighborhood:When α and β are the member of SN, If there is a equation $\exists$g that is member of G and β is a substitution of α with g, we call α as being in the neighborhood of β. We write $\alpha \leftrightarrow \beta$.

Equivalence:When α and β are the member of SN, $\exists\alpha_1, \exists\alpha_2, \ldots, \exists\alpha_n$ are members of of SN, and where $\alpha \leftrightarrow \alpha_1$, $\alpha_1 \leftrightarrow \alpha_2$, ..., $\alpha_{n-1} \leftrightarrow \alpha_n$, $\alpha_n \leftrightarrow \beta$, then we write $\alpha \equiv_N \beta$

Our problem is to compute whether or not $\alpha \equiv_N \beta$ for given N, G, C. The important limitation is that

search space is finite. If there is no limitation to the length, the problem is called word problem. The word problem is known to be undecidable. This means there is no algorithm that always stops. Our problem is decidable since it is a coloring problem of finite set. We might do as follows.

Suppose each of α and β is an element of SN,
1. Let S be { α }.
2. Do following while S grows.
   For all δ that are elements of S, get the neighborhood of δ. Then add to S.
3. If β is a member of S, then $\alpha \equiv_N \beta$, otherwise not $\alpha \equiv_N \beta$.

SN is determined by N and C. Since C is finite, SN is also finite and its size is $size(C)^N$.
Since S always grows and S is a subset of a finite set (SN), theoretically, this procedure always stops. It is, however, not usable in the practical sense because S may become too large to be stored in memory. Although the word problem on finite sets may not be theoretically interesting, it is still a challenge to compute it using actual computers.

3. The Order

The order of strings specifies Critical-Pair/Completion procedure. It is natural to convert the strings into simple form. For example, if we have equation set { "abc"="a" }, we will replace all of "abc" with "a", and not "a" with "abc". The order is used to compare strings, and thus determines the meaning of *simple*. We should carefully choose the order so that the conversion will always stop.

The order that we chose is as follows:
(1) If the two strings differ in length, the shorter string is simpler.
(2) If both strings are identical, neither is simpler.
(3) If the two strings are identical in length, compare the first character using the corresponding integer value. The smaller one is simpler.
(4) Otherwise, compare the strings that begin with the second character.

For the given rule w1->w2, where w2 is simpler than w1 by this definition, we call that *simplifying*. If all rules are simplifying, the transformation result will always be simpler than the original form. Because there is the simplest string (null string), the transformation will stop.

4. The Critical-Pair

Critical-Pair is a pair of strings generated from transformation rules. Both of the strings are equivalent under the original equations, but they have different transformation results. For example, if we have a rule set, {"x"->"b", "x"->"c"}, the pair ("b", "c") is a critical pair. Critical-Pair/Completion procedure generates the additional rule "c"->"b" so that the equivalent strings will always become the same in the final form.

There is a more complex case. For example, if we have the rule set {"ab"->"x", "bc"->"y"}, the pair ("xc", "ay") are equivalent because "xc" = "abc" = "ay". Since both "xc" and "ay" are different transformation results of "abc", the pair ("xc", "ay") is a critical pair of {"ab"->"x", "bc"->"y"}.

The following procedure generates the critical pair of length bounded strings:

Suppose we have rules r1 and r2, where r1 is written as w1->w2, and r2 is written as w3->w4.
(1) Find all of $w \in SN$,
where $w=\alpha\chi\beta$, $\chi \neq ""$, and $(w1, w3) \in \{ (\alpha\chi, \chi\beta), (\alpha\chi\beta, \chi), (\chi\beta, \alpha\chi), (\chi, \alpha\chi\beta)\}$
(2) If such w does not exist, the result is empty.
(3) For each w, found in step 1,transform w with r1 and r2. Let the result be w5 and w6 respectively, add the pair (w5, w6) to the result.

Though this procedure generates all of the Critical-Pairs, some of them are unnecessary for the bounded length problem. Although we do not generates the pair if w is longer than N, there may be a chance where equivalence relation may have intermediate string that is longer than N. As the result, the generated rule may have unnecessary relations where intermediate string are longer than N. Nevertheless, it is important that w is a member of SN. This implies that the length of w is bounded. Without this limitation, the Critical-Pair / Completion procedure may not stop.

5. The Procedure

Although the order and the Critical-Pair may differ, the completion procedure is the same among various problems. The correctness of this procedure is described by Huet [Huet 81]. The following is a description of Critical-Pair/Completion procedure.

Let R be a set of rules, G be a set of equivalence equations.
Set G as initial equations, and R as an empty set.
Do the following:
(0) If G is empty, then R is the obtained rule sets.
(1) Select one element w1=w2 from G and remove it from G.
(2) If w1 and w2 are identical, go to (0).
(3) If w1 is simpler than w2, swap w1 and w2.
(4) Let r be w1->w2.
(5) For all elements w3->w4 of R, if w3 is transformable by r, remove it from R and add w3=w4 into G.
(6) For all elements w3->w4 of R, generate all the Critical-Pair's {(w5, w6)...} from w1->w2 and w3->w4. Add { w5=w6, ...} into G.
(7) Add w1->w2 into R.
(8) For all elements w3->w4 of R, transform w4 into w5 using R, and then replace the rule as w3->w5.

(9) For all elements w3=w4 of G, transform w3 and w4 into w5 and w6 using R, and then replace the equation as w5=w6

G:{"abc"="ab",	"abc"="bc"},	R:{}	; loop 0, step 0
G:{	"abc"="bc"},	R:{"abc"->"ab"}	; loop 0, step 8
G:{	"ab"="bc"} ,	R:{"abc"->"ab"}	; loop 1, step 0
G:{"abc"="ab"},		R:{"bc"->"ab"}	; loop 1, step 8
G:{"aab"="ab"},		R:{"bc"->"ab"}	; loop 2, step 0
G:{"aaab"="ab"},		R:{"bc"->"ab", "aab"->"ab"}	; loop 2, step 8
G:{"ab"="ab"},		R:{"bc"->"ab", "aab"->"ab"}	; loop 3, step 0
G:{}		R:{"bc"->"ab", "aab"->"ab"}	; loop 4, step 0

Fig. 1 : An execution trace.

Steps from (0) to (7) convert one equation to one rule, and add some equation so that the information may not be lost. Steps(8) and (9) simplify the R and G without changing equivalence relations. Fig 1 illustrates the process of procedure for {"abc"="ab", "abc"="bc"} and N > 3.

This procedure stops if steps(6)-(9) do not generate new relation. This means that the procedure either stops or generates new equations. Since the number of equation is finite, the number of all the possible Critical-Pairs is always finite. If this procedure generate enough Critical-Pairs, it will stop generating equations. Then the procedure will stop.

6. String Comparison

For the given equation set G and integer N, this procedure generates corresponding rules R. To compare two strings w1 and w2, first transform both of them using R, then compare the results. If they are not identical, w1 and w2 are not equivalent. If results are identical, w1 and w2 are equivalent for unbounded string domains. However, w1 and w2 may or may not be identical in bounded length.

Suppose, G is {"abc"="ab", "abc"="bc"},N is 3, w1is "aab" and w2 is "abc". This procedure will generate R:{"bc"->"ab", "aab"->"ab"}, then w1 and w2 are transformed into same string "ab". However "aab" and "abc" are not equivalent when N is 3. If N is greater than 3, "aab" and "abc" are equivalent since "aab" = "aabc" = "abc"; however if N is 3, "aabc" is not member is SN because its length is 4.

This means that even the translated results are identical, the original strings may not be equivalent in bounded length. Fortunately, this does not cause any problem in actual application because they are equivalent if the length is not limited.

7. Application

This comparison is applicable when the information is translated from other languages into alphabetical form. For example, if some data comes from Japanese, the name "Ito", "Itou" and "Itoo" may be the same name. Japanese language has a simpler phonetic system than English. As the result, {"chi"="ti", "ci"="si", "shi"="si", "fu"="hu", "fa"="fua", ...} are reasonable equations. Fig. 2 illustrates one example of the Japanese sound system and Fig. 3 shows the obtained rules.

It is interesting to note that original equations has two concepts -- case insensitive and Japanese sound system. The generated procedure satisfies both concepts. It is not always an easy task to mix two equivalence policies at the same time. Since equivalence policy is described in equations, it can be mixed easily in this framework.

```
{"cha"="tya", "chi"="ti", "texi"="ti", "chu"="tyu", "che"="tixe", "cho"="tyo",
"sha"="sya", "shi"="si", "suxi"="si", "shu"="syu", "she"="sixe", "sho"="syo",
"ja"="jya", "ji"="zi", "zuxi"="zi", "ju"="jyu", "je"="zixe", "jo"="jyo", "fa"="huxa",
"fi"="huxi", "fu"="hu", "fe"="huxe", "fo"="fuxo", "ky"="kiy", "sy"="siy", "ty"="tiy",
"ny"=niy", "hy"="hiy", "my"="miy", "ry"="riy", "gy"="giy", "dy"="diy", "by"="biy",
"py"="piy", "a-"="a", "i-"="i", "u-"="u", "e-"="e", "o"="ou", "xa"="a", "xi"="i",
"xu"="u", "xe"="e", "xo"="o", "nn"="n", "xn"="n", "A"="a", "B"="b", "C"="c",
"D"="d", "E"="e", "F"="f", "G"="g", "H"="h", "I"="i", "J"="j", "K"="k", "L"="l",
"M"="m", "N"="n", "O"="o", "P"="p", "Q"="q", "R"="r", "S"="s", "T"="t", "U"="u",
"V"="v", "W"="w", "X"="x", "Y"="y", "Z"="z" }
```

Fig. 2: equations for Japanese pronounciation

```
{"a"->"A", "b"->"B", "c"->"C", "d"->"D", "e"->"E", "f"->"F", "g"->"G", "h"->"H",
"i"->"I", "j"->"J", "k"->"K", "l"->"L", "m"->"M", "n"->"N", "o"->"O", "p"->"P", "q"->"Q",
"r"->"R", "s"->"S", "t"->"T", "u"->"U", "v"->"V", "w"->"W", "x"->"X", "y"->"Y",
"z"->"Z", "A-"->"A", "E-"->"E", "HU"->"FU", "I-"->"I", "NN"->"N", "O-"->"O",
"OU"->"O", "U-"->"U","XA"->"A", "XE"->"E", "XI"->"I", "XA"->"A", "XE"->"E",
"XI"->"I", "XN"->"N", "XO"->"O", "XU"->"U", "ZI"->"JI", "BIY"->"BY", "CHI"->TI",
"CHY"->"TI", "CHY"->"TY", "DIY"->"DY", "FUA"->"FA", "FUE"->"FE", "FUI"->"FI",
"FUO"->"FO", "GIY"->"GY", "HIY"->"HY", "JIE"->"JE", "JYA"->"JA", "JYO"->"JO",
"JYU"->"JU", "KIY"->"KY", "MIY"->"MY", "NIY"->"NY", "PIY"->"PY", "RIY"->"RY",
"SHI"->"SI", "SHY"->"SY", "SIE"->"SHE", "SIY"->"SY", "SUI"->"SI", "SYA"->"SHA",
"SYO"->"SHO", "SYU"->"SFU", "TEI"->"TI", "TIE"->"CHE", "TYA"->"CHA",
"TYO"->"CHO", "TYU"->"CFU", "JUI"->"JI"}
```

Fig. 3: transformation rules for Japanese pronounciation

8. The Importance

Small data variances cause the failure of information retrieval. This variance frequently happens when original data comes from non-English languages. The Japanese name is a clear example. If the lanaguge has non-alphabetical writing, the mapping between original character to alphabet form may not be one-to-one mapping. The word problem is a practical problem in this situation. Although superfluous data may be equivalent, we can reject this data after the selection. The failure of retrival is more problematic than superfluous output.

9. Related Works and Future Works

Knuth [Knuth 70] introduced this procedure and applied it to term-rewriting-system. Buchberger [Buchberger 87] surveyed the overall characteristics of Critical-Pair/Completion procedure. Book [Book 87] explained the relationship between string and term-rewriting-system. This paper is based on these works. Our contribution is to find the problem that is simple enough to be always solved, and that is powerful enough to be applied actual information systems.

We are applying these conversion rules to pick up telephone directory by names. People's name may have data specific equations. For example, "kamihayashi", and "kambayashi" may be the same person because the sound is changed due to the combination of "kami" and "hayashi". These strings are equation candidates in actual data retrieval. In this case, the number of equations is in the thousands.

In the future, it will be important to calculate computational complexity. Although we briefly explained that the procedure will stop, the worst case is still the exponential order of N. This would be no better than simple coloring procedures although our experience shows that our procedure actually ends in reasonable amounts of time.

10. Conclusion

This paper formulated Critical-Pair/Completion procedure for bounded length strings. This procedure generates rewriting rules from substring equations. We can test string equivalence under certain equations using these rules. This paper also explains that the string equivalence is an important problem for international data handling.

Acknowledgement

The author thanks Yoshihito Toyama(JAIST) and Hirofumi Katsuno (NTT) for his discussion and guidance about Critical-Pair / Completion procedure. The author also thanks Katsumi Takahashi (NTT) for his discussion about actual information system such as telephone directory services.

References

[Book 87] R. V. Book, "Thue Systems as Rewriting Systems," Journal of Symbolic Computation Vol.3, No.1, pp39-68, 1987

[Buchberger 87] B. Buchberger, "History and Basic Feature of the Critical-Pair/Completion Procedure," Journal of Symbolic Computation Vol.3, No.1, pp1-38, 1987

[Huet 80] G. Huet, "Confluent Reductions: Abstract Properties and Application to Term Rewriting Systems," Journal of the ACM, Voll.27, No.4, pp.797-821, 1980

[Huet81] G. Huet, "A complete proof of correctness of the Knuth-Bendix completion algorithm," Journal of Computer and System Science, Vol.23, No.1, pp.11-21, 1981

[Squire87] C. Squire and F. Otto "The Word Problem for finitely presented Monoids and Finite Cannonical Rewriting Systems.", Lecture Notes in Computer Science Vol.256, pp74-82, 1987

BILINGUAL ALIGNMENT AND TENSE

Diana Santos

INESC
R.Alves Redol, 9, Apartado 13069, P-1000 Lisboa, Portugal
dms@inesc.pt

Abstract

In this paper, I describe one annotation of tense transfer in parallel English and Portuguese texts. Even though the primary aim of the study is to compare the tense and aspect systems of the two languages, it also raises some questions as far as bilingual alignment in general is concerned. First, I present a detailed list of clausal mismatches, which shows that intra-sentential alignment is not an easy task. Subsequently, I present a detailed quantitative description of the translation pairs found and discuss some possible conclusions for the translation of tense. Finally, I discuss some theoretical problems related to translation.

1. Introduction

Many people have recently suggested that aligned texts could be used for several NLP tasks such as automatic building of bilingual terminology, or machine translation (see the Call for Papers for this workshop). In this paper, I describe a study conducted on manually aligned bilingual corpora, at the clause level, annotated with respect to tense transfers.

This study shows that, given sentence alignment, it is not a simple task to proceed to smaller units: In fact, there is considerable divergence from the desirable one-to-one clause correspondence. In addition, and as a possible morphological clue, tense is not a reliable indicator for a more refined alignment (Section 2).

On the other hand, it provides the first quantitative results of tense translation that I am aware of, and the way those results may be used is discussed (Section 3). However, one should not forget that these empirical results should be evaluated in the light of a theory of translation. Without stating one's view on the semantic relation between translations, one cannot draw any conclusions from parallel texts (Section 4).

2. The study

I selected an (American) English novel[1] and its translation into (European) Portuguese and a collection of Portuguese short stories[2] and their translation into (American) English. The type of the texts is narrative discourse, thus providing a fertile ground for studying the translation of tense and aspect. Given that the texts are relatively small, they are characterized by favouring the description of actions. In addition, none of the original texts can be considered "difficult" by the ordinary reader. A relatively detailed quantitative description of the texts is given in Tables 1 and 2.

English original	Words in English	Words in Portuguese	Sentences (Eng)	Sentences (Port)	Transl pairs	Tensed transl
EP1	3416	3051	211	224	207	535
EP2	2233	2017	115	127	113	248
EP3	5815	5172	349	410	340	782
EP4	5051	4554	347	393	337	701
EP5	2827	2472	194	235	191	396
EP6	6718	5996	412	472	412	1082
Total	26060	23262	1628	1861	1602	3744

Table 1

Portuguese original	Words in Portuguese	Words in English	Sentences (Port)	Sentences (Eng)	Transl pairs	Tensed transl
PE10	4410	4898	323	324	322	596
PE11	1501	1719	70	71	70	217
PE3	3447	3695	210	211	210	456
PE8	2019	2279	107	107	106	239
PE6	4460	5258	352	355	351	595
PE9	4393	4698	208	210	208	487
Total	20230	22547	1270	1278	1267	2590

Table 2

The texts were aligned manually, at sentence level. As is well known, some cases X-Y occurred where both X and Y contained more than one sentence. I numbered these "translation pairs".

Sentence alignment numbers follow. They were actually negligible for the Portuguese originals. I should also note that a considerable proportion, in the case of the English original, came from direct speech conventions, completely different in the two languages.

Source-target	PE10	PE11	PE3	PE8	PE6	PE9	Total
1 - 2	2	1	1	1	4	2	11
2 - 1	1			1	1		3

Source-target	EP1	EP2	EP3	EP4	EP5	EP6	Total
1 - 2	16	12	46	32	20	48	174
2 - 1	3	2	4	2	3		14
1 - 3		1	8	9	8	6	32
2 - 2	1		1	2	3	1	8
2 - 3			1	1	1		3
3 - 1				1			1
3 - 2				1			1
3 - 3			1			1	2
2 - 4				1			1
1 - 4			1		1		2
1 - 5					1		1

Then I typed the tense transfer for each clause. It is important to understand what I mean by "clause" here. Since I am specifically interested in tensed clauses, untensed ones were only counted if they were the translation of tensed ones. Clause-like structures without a verb have not been counted as clauses, but VP conjunction is counted as two clauses.[3] Conversely, when two or more verbs are rendered by only one, I have only counted **one** transfer. This means that the number reported under "Tensed transl" above is less than the number of tensed clauses in either of the two languages.

Let me now describe the annotation in detail. First, throughout the paper I use "tense" for tense forms which can be simple or complex: for example, present progressive is considered a tense distinct from simple present. I generally omit the word "simple", too. The annotations were kept in separate files, in the following format: For each translation pair (resulting from sentence alignment), I recorded: First, the "translation pair" number, then the tense transfer, preceded by the number of times if consecutive. The tense transfer is displayed by a mnemonic for the source tense, a dash, and a mnemonic for the target tense, or, if the mnemonic is shared by the two languages, it appears only once (so *pres*, instead of *pres-pres*). Tense transfers in each translation pair are separated by semicolons. In case the main source verb is *be*, *ser* or *estar* (the two usual translations of *be*), the verb precedes the specification of the tense transfer (e.g. *3 x be PS - I*). One example is translation pair 273 of EP4:

> *4.273 His senses were burningly alive, but his mind went back to the deep participation with all things, the gift he had from his people.*
>
> *Os seus sentidos ardiam, vivos, mas, com aquele dom que os antepassados lhe tinham transmitido, regressava à íntima comunhão com todas as coisas.*

which is annotated:

273	be PS - I; PS - I; PS - MQP;

This example was chosen because a) it shows a "be" main verb which does not get translated by a corresponding Portuguese "be" (and that is not marked in the annotation); b) there is a clause inversion: C1 C2 C3 in English is rendered as TC1 TC3 TC2 in Portuguese.

During the course of this annotation, my views about **clause** alignment changed. *Prima facie*, it should be more reliable than sentence alignment, since the division in sentences seemed more a stylistic matter than a semantic one: E.g., it seemed less harmful to translate a two-claused sentence into two sentences of one clause each (or vice versa) than to change the overall number of clauses. However, contrary to my intuitions, I found out that the number of clause mismatches even outnumbers that of sentence mismatches.

Here, I systematize and illustrate clause misalignment by means of a set of examples, presented in order of decreasing predictability. (The labels given do not carry any claims as to theoretical analysis of the phenomena described). If this were the main purpose of the study, I should have separated the English to Portuguese mismatches from those in the opposite direction. However, being only a by-product of the main investigation, and given space limitations, I grouped them together.

1. Differences (well?) known from the grammars of the two languages.

a) Dependent clauses which are tensed in one language and not in the other

10.119 *-- Ajuda-me a despendurá-lo, Marco Semprónio. Eu quero que ele **viva**.*
*"Help me unfasten him, Marcus Sempronius, I want him **to live**."*

b) Temporal clauses that may have different tense requirements

10.110 *-- Que foi que ele revelou **antes de desfalecer**?*
*"What did he reveal **before** he **lost** consciousness ? "*
5.64 *We must be gone **before** the daylight **comes**."*
*Temos de fugir **antes que** o dia **nasça**.*

c) Subbordinated clauses attached to noun phrases

11.20 *Não era uma tentação que repelia assim; mas era, como bem sabia, um esforço **para que** o céu se **contentasse** com as relações espirituais de uma oração.*
*It was not a temptation that she repelled in this way; but it was, as she well knew, an effort **to satisfy** the heavens with the spiritual offering of a prayer.*

2. Translation between tensed and non-tensed clauses

a) Main into adverbial

11.51 *É certo que, por mais que fizesse, ocasiões havia em que se afastavam*

dela as outras, a ***deixavam*** *só, como se a propiciarem a repetição de acontecimentos que eram honra do convento.*

For invariably, whatever they did, there were moments when the others would go away from her, ***leaving*** *her alone, as it to propitiate the repetition of events that were the honor of the convent.*

b) Main into adnominal

5.17 He turned away from her and walked up the beach and through the brush line. His senses ***were dulled*** *by his emotion.*

Voltou as costas à mulher, subiu pelo areal, atravessou as sebes, com os sentidos ***embotados*** *pela emoção.*

c) Relative into gerundive

10.11 -- Ouviram uma voz ***que gritava****, não* ***gritava****, não, mas* ***soluçava****,* ***uivava****, era um rugido triste, dentro da noite, em cima do cabo, ou dentro dele...*

"They heard a voice ***shouting****, no, not* ***shouting****, but* ***crying*** *out,* ***howling****, it was a sad wailing, deep in the night, above the cape, or from within it...."*

d) Infinitive into main

10.208 Há sempre quem suponha, na sua paixão, que destruir Roma, a devassa Roma, a pecadora Roma, é ***dar*** *testemunho dos desígnios de Deus.*

There are always those who, in their zeal, believe that by destroying Rome, that debauched Rome, that sinful Rome, they ***are carrying*** *out the designs of God.*

3. Reduction of the overall number of clauses

a) Conjunction reduction

5.35 Her back ***was*** *bent with pain* ***and*** *her head* ***was*** *low.*
Levava *as costas curvadas pela dor* ***e*** *a cabeça caída.*

b) Conjunction merging into a complex predicate

*5.123 "****Go*** *now to Juan Tomás* ***and bring*** *him here and tell no one else.*
Vai buscar *João Tomás e não digas nada a mais ninguém.*

6.115 She took him up quickly and put him under her shawl and ***gave*** *him her breast* ***and*** *he* ***was*** *silent.*

Joana agarrou nele bruscamente, pô-lo debaixo do xale, ***fê-lo calar*** *com o seio.*

c) Adverbial into adjectival modification

6.202 And the path ***rose steeply*** *now, so that he panted a little as he went.*
O caminho, agora ***escarpado****, fazia Kino arquejar na subida.*

d) Clause into prepositional or noun phrase (see also 6.202 above: "as he went" into "na subida")

6.36 But Kino sat on the ground and ***stared*** *at the earth in front of him.*
Mas Kino ficou sentado no chão, ***com os olhos*** *na terra.*

e) Deletion of perception clauses

6.7 Kino could feel the blown sand against his ankles and he was glad, for ***he knew*** *there would be no tracks.*

Sentia com alegria a areia fustigar-lhe os tornozelos, porque, assim, não deixariam pegadas.

4. Expansion of the overall number of clauses

a) Adding perception clauses

6.170 *His work would come last, for he would not take them back.*
O seu trabalho ficaria para o fim porque não ***pensava*** *levá-los para trás.*

b) Adding verbs to sentences with no verb

6.67 *And there in the pearl Coyotito' s face, thick and feverish from the medicine.*
E, na pérola, ***viu*** *o rosto de Coyotito inchado e febril com o medicamento.*

b) Conjunction expansion

5.34 *The pale moon* ***dipped in and out*** *of the strands of clouds so that Juana walked in darkness for a moment and in light the next.*
A pálida Lua ***mergulhava e emergia*** *desses fiapos de nuvens. E Joana caminhava, ora às escuras, ora iluminada.*

5. Simply no correspondence

a) Ommission

6.213 *The animals from miles around came to drink from the little pools,* ***and the wild sheep and the deer, the pumas and raccoons, and the mice--all came to drink.***
De quilómetros em volta, os animais vinham beber àquelas lagoas.

b) Creation

6.284 *"If they kill me," he said, "lie quietly.*
Acabou por *dizer: --Se eles me matarem, não te mexas.*

This last item (point 5) is connected to the first problem I had to deal with: the existence of blatant mistranslations[4]. To my astonishment, they[5] were fairly frequent. Surely, not all cases can be attributed to the translator alone. Some could have originated in the use of a slightly different version of the original text, with typos, or else could themselves be typos in the final published version. For this study, I decided nevertheless to count the temporal information of those examples where the mistake lay somewhere else.

Summing up, this is a first (and surely incomplete) survey of tensed clause mismatches. Since I was primarily concerned with tense, I did not mark the number mismatch in the annotation, except for the case of tensed - untensed clause. So, I was only able to show a qualitative description of the phenomenon. My impression is nevertheless that non-agreement of clause alignment was frequent[6]. In addition, I have also not marked the cases where actual changes of order among the translated clauses occurred, which is undoubtedly another complicating factor for automatic alignment (cf. the example 4.273 above).

3. Survey of actual translation pairs

Before discussing the results, I should explain why this study was conducted.

First, I believe that language studies should rely on corpora and not on subjective considerations, since language is not what we would like it to be. In addition, relative frequencies of problems and of their theoretical discussion in the literature are completely unrelated, to the extent that only a superficial examination of the data can already provide interesting information.

Therefore, I wanted not only to find regularities in tense transfer but to see the problems as well. For example, superficial mismatches (such as theoretically unforeseen pairs, or clause number mismatches) could be a good pointer for problematic transfer cases.

I turn now to these two points in turn, after presenting a quantitative description of all cases of tensed clauses translated into another clause, tensed or untensed. For lack of space, only data concerning the past tenses are shown: In upper case, the source tense, then the names of the target tenses used in the translation. The cases of main verb "be" are shown in parentheses.

	EP1	EP2	EP3	EP4	EP5	EP6	Total
PAST SIMPLE	323 (54)	185 (39)	499 (52)	412 (69)	242 (33)	644 (85)	2305 (332)
imperfeito	125 (38)	85 (34)	175 (34)	179 (56)	89 (29)	260 (73)	913 (264)
perfeito	161 (9)	75	265 (8)	192 (7)	123 (2)	318 (7)	1134 (33)
infinitivo	5	4	10 (2)	9	9	20	57 (2)
gerúndio	5	2	10	3	6	9 (1)	35 (1)
imperfeito conjuntivo	3		3	10		10 (1)	26 (1)
mais-que-perfeito	6 (2)	5 (3)	11 (2)	10 (2)	4 (1)	3 (1)	39 (11)
condicional	1 (1)		4	2	4	3 (1)	14 (2)
presente		5 (1)	5	1 (1)		4	15 (2)
presente conjuntivo		1			1	3 (1)	5 (1)
part. pass.	4 (2)	3	3	5		3	18 (2)
ir gerúndio	2	2	4	1	2	1	12
pôs-se a infinitivo	2	2		1		5	10
estar part. passado	2		1	1	1	1	6

For the simple past, only forms which have been used more than five times are listed. It has also been translated by a prepositional phrase, a noun phrase, periphrases employing Portuguese 'began' 'made', 'got', etc..

	EP1	EP2	EP3	EP4	EP5	EP6	Total
PAST PROG	8	5	19	4	8	14	58
imperfeito	3	4	12		5	8	32
imperfeito prog.	1	1	2	3	2	4	13
gerúndio			2			1	3
mais-que-perfeito	2				1		3
ir ger	2		1				3

For the past progressive, I used a threshold of two. Once, it was translated by infinitivo, by presente, by an adjective, etc..

	PE10	PE11	PE3	PE8	PE6	PE9	Total
IMPERFEITO	127	109	155	99	198	200	888
past simple	101	91	120	64	136	156	476
past progressive	11	5	8	16	20	11	71
gerund	5	3	2	3	3	6	22
could	3	2	4	4	4	7	24
conditional	1	1	8	1	15	4	30
used to	2		1	2			5
passive			1	4	4	4	13
pluperfect			1	3	5	4	13
infinitive			2				2

For the Portuguese tenses, I listed all cases translated more than once. Imperfeito was once rendered by a past simple in the passive voice, and by *came to* infinitive. All occurrences of *ser* and *estar* in imperfeito were rendered by simple past in English.

	PE10	PE11	PE3	PE8	PE6	PE9	Total
PERFEITO	229	37	109	79	108	74	636
past simple	201	35	100	70	100	69	575
present perfect	13		3	2	5	2	25
present	4			1			5
passive	2	1	2	2	1		8
gerund	2					1	3
could			1		3	2	6
pluperfect				3	1		4
conditional	2						2

The other perfeitos were translated once by the periphrastic expressions '*stopped* plus gerund' and '*went* plus gerund'.

These numbers demonstrate that there is far from a one-to-one correspondence between tenses. This is clearly not an original statement, but still one that has not, to my knowledge, been formally documented, nor taken into account as a performance problem by practical systems. Given the results above, it seems fair to say that one cannot rely on tense indications for automatic alignment of clauses, since even the trichotomy past, present and future is not preserved in general.

I will try nevertheless to draw some first conclusions[7]:

1. The low frequency of progressive in English[8] is most surprising, even though it is known that it is more frequent in speech than in writing. This casts doubt on the traditional (Kamp, 1981) DRT analysis of progressive filling the same role as French imparfait (by itself and also if we assume that Imperfeito is similar, to a great extent, to the French tense).

2. It seems that the distinction between foreground and background in English is not given primarily by tense, since the vast majority of tenses (62%) were past simple. This agrees with Couper-Kuhlen's contention, (1987:24): "*(...) for the organization of temporal relations in narration the contribution of the Past tense (as opposed to some other tense) is minimal (...)*".

3. Couper-Kuhlen (1987, 1989)'s description of relative clauses "not advancing narrative" was also supported by this study: in all relative clauses with a relative pronoun in EP1 and EP2, 39, only one (2.6 %) was translated by a perfeito.

4. Stativity in English strongly favoured imperfeito. But this is not an easy thing to determine. In fact, the easiest clue (main verb "be") was only translated by Imperfeito in 80% of the cases.

5. As far as Portuguese to English is concerned, only a negligible fraction of imperfeitos (4%) was translated by conditional or periphrastic "used to". In order to check my own contention (Santos, 1993) that habituality is an important feature of imperfeito; I analysed every occurrence of this form in PE10, concluding that 12 imperfeitos (in 127) could be considered habitual. The vast majority of the imperfeitos was used in connection with verbs of being (32), cognition, perception and locatives. Imperfeito was also frequently used to describe an extended action in progress, in 21 cases.

6. While the use of imperfeito to convey "the action in progress" is observed in some translations into the English progressive, no tense difference is observable in the English translation for the most part (actually in 15 of the above 21). In fact, progressive was used for other reasons in the other 5 cases.

These two last observations show a less explicit tense system in English as far as the dimensions habituality/non habituality and in progress/finished are concerned.

On the hope of getting evidence that an automatic tool could help finding mistakes in translation, the answer seems to be negative. On one hand, it was clear that superficial mismatches do not necessarily mean mistranslations, as can be checked in the examples presented above. On the other hand, I was able to find several cases where the tense translation was common although I believed the translation was incorrect. Cf. the following translation from simple past into imperfeito, where perfeito should have been chosen:

5.49 His senses were coming back and he ***moaned****: "They have taken the pearl..."*

Ele estava a voltar a si, ***gemia****.--Roubaram a pérola...*
(He was coming back to himself, he moaned (imperf): -)

Still, one could gather some complex bilingual dictionary entries in the cases where the translation mismatches were due to lexical matters. What was left unproved, at best, was the possibility of

building a translation checker program working on unrestricted aligned text in the two languages.

4. Theoretical considerations

The empirical studies and tentative results described so far have, I believe, to be put in perspective as far as the theoretical issue of translation equivalence is concerned.

In the first place, one has to clarify the ontological status of tense: Tense can be seen as a purely grammatical category, and the grammars of two different languages would not require any similarity (as is the case with grammatical gender, for example). This has been recently argued for by Vlach (1993), who claims that tense is an idiosyncratic feature which does not carry meaning outside the grammatical system of one particular language.

But tense can be -- and generally is -- regarded as having a strong semantic import, together with an important discourse role. If semantic content and discourse function are preserved by translation, then it is puzzling why in the material above tenses diverge so much.

However, I believe that translation does not preserve the meaning conveyed by a text (even though it ideally approximates it). This claim is fairly old and is held by a multitude of researchers in the translation (Bar-On, 1993) and philosophical camps (Keenan, 1978), even though typically researchers in machine translation (as well as in tense and aspect studies in general) continue for the most part to presuppose the equality of meanings across languages.

If translation does not necessarily preserve meaning, one should be very careful in comparing translation pairs, since if, in some cases, the meaning can be regarded as the same, in others, it has to be seen as an optimal aproximation (or less). For actual examples of these "semantic mismatches", see Santos (1994).

Thus, all translation pairs do not have the same status. In order to use them meaningfully, one needs to assess first what descriptive content and kind of use can be sactisfactorily rendered by a pair of languages, then study separately the cases where this is not the case, i.e., when the two languages have different expressive power, or efficiency requirements (Keenan, 1978).

Before drawing substantive conclusions, one should make the distinction among

1. those cases where there is superficial disagreement but a semantically satisfying translation;
2. those cases where there is semantic disagreement because of the characteristics of the two language systems;
3. and finally those cases while there is semantic disagreement as a translation option[9].

Putting the three cases in the same bag only obscurs the whole issue of translation instead of clarifying it.

5. Practical conclusions

- The number of tensed verbs in the two languages is not a reliable indicator for sentence alignment.
- Robust clause alignment does not seem to be feasible using tense indicators in the two languages. Surely, this point is only of importance if one expected to perform alignment based on unilingual analyses, that is, having an annotated English and an annotated Portuguese text but no specific contrastive resources.
- The size and the non-balancing of the texts used does not allow to generalize as to specific probabilities of translation between English and Portuguese in general. However, given that the number of source tense forms is fairly small, it seems right to assume that some sort of Zipf's law holds for tense transfers: Given enough data, the probability of the strangest tense translation is greater than zero. In other words, if we annotated more and more narrative text we would get more and more instances of the translations pairs already found, in roughly the same proportion, while some new (and rare) combinations would still be found.
- A corpus study is important for eliciting some regularities in the translation of tenses and also to discover factors that may play a role in it.
- Tense translation embodies different degrees of "correctness", which should be studied before one can draw conclusions about specific pairs of tense forms. In particular, worse than the difficulties of clause alignment and of tense translation is the fact that actual translations may not preserve the information content of the original text and may add information which is not there.
- Aligned texts are not so obviously fit to use in practical systems. One has to perform in-depth studies of various subjects in order to be able to use them usefully.

Acknowledgements

I gratefully acknowledge a PhD grant from Junta Nacional de Investigação Científica, and am grateful to Lauri Carlson for his supervision and detailed comments, and to Jan Engh for helping me to make some points clearer.

References

Bar-On, Dorit. "Indeterminacy of Translation : Theory and Practice", *Philosophy and Phenomenological Research*, Vol. LIII, No. 4, December 1993.

Couper-Kuhlen, Elizabeth. "Temporal relations and reference time in narrative discourse", in Schopf, Alfred (ed.), *Essays on Tensing in English. Vol 1: Reference Time, Tense and Adverbs*, Niemeyer, 1987, 7-25.

Couper-Kuhlen, Elizabeth. "Foregrounding and temporal relations in narrative discourse", in Schopf, Alfred (ed.), *Essays on Tensing in English. Vol 2: Time, Text and Modality*, Niemeyer, 1989, 7-29.

Kamp, Hans. "Evènements, représentations discursives et référence temporelle", *Language 64,* 1981, 39-64.

Keenan, Edward L. "Some Logical Problems in Translation", in F. Guenthner & M. Guenthner-Reutter (eds.), *Meaning and Translation: Philosophical and Linguistic Approaches*, Duckworth, 1978, 157-89.

Santos, Diana. "Integrating tense, aspect and genericity", *Actas do IX Encontro da Associação Portuguesa de Linguística* (Coimbra, 29/9-1/8/93), 391-405.

Santos, Diana. "Translation mismatches and tense and aspect", 1994, submitted to *Computational Linguistics*.

Vlach, Frank. "Temporal Adverbials, Tenses and the Perfect", *Linguistics and Philosophy 16,* 1993, 231-83.

---

[1] *The Pearl,* by John Steinbeck. Each chapter will be described by EPn, from 1 to 6.

[2] By Jorge de Sena, respectively: PE10: *A noite que fora de Natal* (A Night of Nativity); PE11: *O grande segredo* (The Great Secret); PE3: *Mar de pedras* (Sea of Stone); PE8: *A campanha da Rússia* (The Russian Campaign); PE6: *A comemoração* (The Commemoration), and PE9: *Kama e o génio* (Kama and the Genie).

[3]Basically, the number of tensed "clauses" is that of tensed verbs (with one proviso: see below), but I talk about clauses because I am interested in comparing tense and aspect which are influenced by all participants in a clause.

[4]Note that (6.213) is not a mistranslation. As pointed out by Lauri Carlson, it is possible that "the Portuguese translator omits listing all the local animals coming to the waterhole because she estimates the effect of the list on her supposed audience would be opposite of its effect on readers of the original --- making the image more concrete and familiar for the former who have intimate acquaintance with the local fauna, but causing puzzlement or estrangement for foreign readers".

[5]I am not including in this category the (rather frequent) cases where I did not agree with the specific choice made by the translator, but it was possible that the published translation was

intended.

[6]Note that this description was only based on the files EP5, EP6, PE10 and PE11, a much smaller corpus.

[7]But acknowledging that these corpora should be used to measure detailedly factors that may influence the selection of a particular tense form in translation, such as kind of argument (definite/indefinite, singular/plural), stativity, previous or following tense forms, matters on which I expect to report soon.

[8]And especially in Portuguese.

[9]Given that human translation is a highly elaborate cognitive task, and thus heavily grounded on intuitions and subjective knowledge of language and world, it is not amenable to description in algorithmic terms, for instance as analysis of all possible renderings and their semantic implications. Therefore, "translation options" may be done without full conscience of the translator. Hence, I suggest that (non-trivial) translation errors should be regarded as options as well.

Comparative Discourse Analysis of Parallel Texts

Pim van der Eijk

Digital Equipment Corporation*

Ratelaar 38

3434 EW Nieuwegein

The Netherlands

eijk@cecamo.enet.dec.com

Abstract

A quantitative representation of discourse structure can be computed by measuring lexical cohesion relations among adjacent blocks of text. These representations have been proposed to deal with sub-topic text segmentation. In a parallel corpus, similar representations can be derived for versions of a text in various languages. These can be used for parallel segmentation and as an alternative measure of text-translation similarity.

*The research reported in this paper was partially sponsored by the European Commission, through the LRE 62-050 project, Multext.

1 Introduction

The study of large collections of texts and their translations has recently received much attention in the field of computational linguistics. In this paper, we discuss a trilingual application of earlier research on quantitative representations of the discourse structure of texts, derived from measurements of lexical cohesion. The representations are computed by measuring the similarity between vector representations of adjacent text segments, following a proposal in (Hearst, 1993). In her paper, the representations are applied to the task of segmenting long texts into sequences of discussions of subtopics, called 'tiles'. Experiments are reported that indicate that the tiles correspond rather well to human judgments on document structure.

In this paper, we apply these representations to documents of which multiple language versions are available. In a reasonably well translated parallel corpus, discourse structure seems to be a foremost property that should be preserved across translation. From the point of view of discourse analysis research, parallel corpora could thus be used profitably as a resource to evaluate and compare text segmentation prototypes. From the point of view of translation research, the correlation between the vectors of similarity measurements can also be used directly as a measure of one component of text translation similarity, that can be used in addition to other measures, such as length of aligned text segments or lexical information. The techniques discussed in this paper could also be used as an alternative knowledge source for tools to align parallel documents.

This paper is structured as follows. First we discuss lexical cohesion, which is used to measure similarity of adjacent pairs of text segments. The similarity measures can be viewed as sample measurements of a 'discourse signal'. We will then briefly discuss the trilingual corpus that we used for experimentation and for evaluation, and the linguistic analysis applied to it. Different language versions of a single document will yield different discourse 'signals'. The similarity of signals can be measured by analyzing the discrete correlation of the representations.

2 Discourse Structure Analysis

The discourse structure of a document is analyzed by tracking patterns of semantically related elements in texts. We will first introduce some terminology, and then discuss how this structure can be computed.

2.1 Cohesion

In a coherent discourse, a text is not a random sequence of sentences, but rather sentences are linked by relations such as elaboration, exemplification, and cause. These relations contribute to the *coherence* of texts. There are no computational mechanisms yet to compute coherence, but it is possible to detect *cohesion*. Cohesion arises from back-references, conjunction, or lexical cohesion. Lexical cohesion is the cohesion that arises from semantic relations among words (Morris and Hirst, 1991). Lexical cohesion can be subdived in a

number of classes, such as reiteration of word forms, reiteration by means of superordinates, and reiteration by means of semantically related words (either or not systematically classifiable).

For our prototype, we only detected cohesion caused by reiteration of word forms, with some provision for morphological variation (cf. section 3). Earlier research has used *Roget's Thesaurus* (Morris and Hirst, 1991) and WordNet (Hearst, 1993) as sources of semantic categorization to help detect cohesion arising from reiteration of distinct words that belong to a single semantic class. Thesaurus classes can be used instead of, or in addition to, lexical index terms. To detect reiteration of morphological variants of a word, dictionaries or morphological analysis tools are needed. For our prototype, we used lemmatizers for English and German derived from lexical lists from Celex[1] to map unambiguous word forms to their lemma forms.

2.2 Computing cohesion

Cohesion relations can be used to compute the similarity of text segments. The basic approach (vector similarity measurements among adjacent pairs of text segments) is the one proposed by Marti Hearst, but some details are different, in particular the linguistic analysis described in section 3 and the choice of the digital filter. To compute similarity, segments are first analyzed as weighted vectors of index terms. Index terms are the word forms or lexemes occurring in

[1]The Celex material is available on CD-ROM from Celex and from the LDC.

the corpus. Term weights are computed using the idf.tf measure commonly used in vector-space approaches to information retrieval (Salton and McGill, 1983). This measure expresses that salience of terms in a segment is proportional to frequency in a segment and inversely proportional to segment frequency. A stoplist of function words can be used to restrict the attention to content words, although they have very low weights and thus only a limited influence on the similarity measure.

To compute the cohesion of a text, the similarity between the vector representations of segments is computed. The vectors can be viewed as points in a multidimensional space, where similar vectors 'point in the same direction', so that similarity can be measured using the cosine of the angle between them:

$$cos(x,y) = \frac{\sum_{t=1}^{n} w_{t,x} w_{t,y}}{\sqrt{\sum_{t=1}^{n} w_{t,x}^2} \sqrt{\sum_{t=1}^{n} w_{t,y}^2}}$$

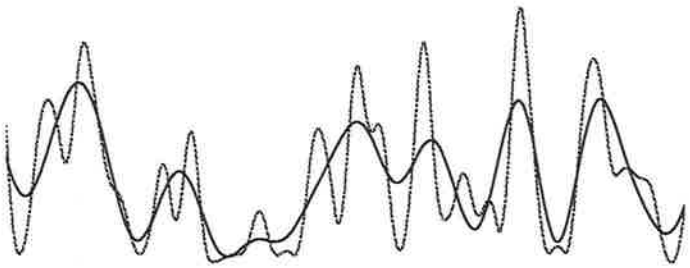
Because terms are weighted using idf.tf, terms that are frequent in both segments, but infrequent in the document as a whole, contribute most to segment similarity. A document is then represented as a vector of cosine values corresponding to the sequence of pairs of adjacent segments, that can be plotted, yielding a wave-like figure. The waveform can be interpreted as follows: increasing values indicate continued discussion of a subtopic. Valleys mark the transition from one subtopic to another. TextTiling divides a text in regions spanning the intervals between minimal values. As discussed in (Hearst, 1993), there is a fairly good correspondence between the tiles marked by TextTiling

and human judgements on text structure.

However, the measurements should be viewed as raw approximations of the discourse structure, because only a subset of cohesion relations are detected, and cohesion is only one factor contributing to coherence. The first strategy to improve on this is to improve linguistic analysis. We applied (cf. section 3) an unsophisticated morphological analysis and no semantic word class information at all. It should be noted that, for the translational applications, it is not necessary to apply a uniform analysis method to all languages. We only compare the similarity measurements, which need not be produced using identical analysis steps. This is an advantage, because in practice one often lacks comparable resources, such as a lexical list or thesaurus, for different languages.

Apart from improving the analysis, a second strategy that can be applied is the use of digital filters to improve the representations. In particular, we used a low-pass filter to smooth the signal, by eliminating high-frequency components in the signal. This operation eliminates small local minima and maxima, and is needed to emphasize the general trends of the graph. To program these and other functions, we used the signal processing functions from the DXML library (DXML, 1993). In the plotted display (figure 1) the smoothed and unsmoothed representations of the English version of the UBS-corpus are displayed.

Figure 1: Effect of lowpass filtering



3 Trilingual Corpus

To evaluate the techniques, we used English, German and French versions of a banking report of the Union des Banques de Suisse (UBS) discussing developments in the Swiss economy in 1987.[2] The texts were analyzed linguistically in a number of ways. First of all, the corpus was aligned at the paragraph level. A very simple lexical analysis was subsequently applied to the three texts.

Paragraph alignment In a first pass, markup was inserted in the texts to identify boundaries of 'segments' (paragraphs and headings). The texts were then semi-automatically aligned at the segment level by first matching headings, based on segment size. Headings can be distinguished from paragraphs rather easily. Small divergences (cases where two paragraphs were mapped to a single (larger) paragraph) were then easily detected, and corrected manually. The

[2]This corpus was kindly made available by Susan Armstrong, ISSCO, Geneva.

resulting corpus consisted of three parallel lists of 484 paragraphs. These are then used to generate three parallel arrays of similarity measurements, for 483 pairs of adjacent paragraphs.

Lexical analysis The three language versions were analyzed lexically by removing numbers and punctiation and by converting all words to lower case. We applied a conservative type of lemmatization to the English and German versions of the document using word lists from Celex, by replacing unambiguous inflected forms by their citation forms. Ambiguous and unknown word forms were therefore left unchanged, and retained as index terms. No stoplist was used. For English, this resulted in a reduction of word form types from 4059 to 3493 for 31518 tokens. The German text contained 6233 word form types and 27019 tokens. The lower number of tokens and higher number of types is due to the productivity of compounding and spelling conventions for compounds[3] and to richer morphology. Lemmatization of unambiguous word forms reduced the number of word form types in German to 5372. Both lemmatization operations eliminate about 14% of the index terms.

The French version of the document contained 4498 word form types and 32805 tokens, and was not lemmatized for lack of a lexical list or morphological analyzer. We did evaluate a crude analysis method taking ngrams of characters

[3]The compounding issue can be shown to be a problem for detection of word correspondences. When dealing with Dutch, which is similar to German in this respect, detection of correspondences was found to be done best at the phrase level rather than the word (form) level (van der Eijk, 1993).

as index terms instead of word forms. This approach was remarkably successful (cf. section 4), so we also applied it to the English and German texts.

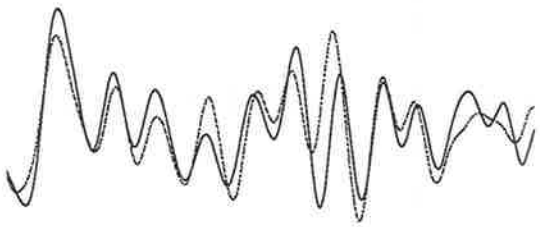
4 Application to parallel corpora

By TextTiling, an attempt is made to capture the implicit semantic structure of a text in terms of a series of subtopics. An interesting way to evaluate and extend the use of these techniques is to apply them to a multilingual corpus.

The subtopic structure can be viewed as a property of the text which should, to some extent, be shared by a text and its translations. The paragraph similarity measurements are ultimately based on repetition of lexical material. These repetitions need not necessarily hold for the parallel segment pairs, e.g. two occurrences of a word might be translated by two synonyms, which will not be recognized as lexical cohesion in the absence of a thesaurus. In our corpus, this problem already arises by the poor morphological analysis. Another problem is noise arising from word sense ambiguity: a term might be used in distict senses, resulting in a spurious case of lexical cohesion.

Nevertheless, we hypothesized that, at the paragraph level, these discrepancies would more or less 'level out', i.e. one divergence might be offset by another convergence. This turned out to be the case, as illustrated graphically in figure 2, for part of the German and English versions, analyzed using character trigram index terms. The overall 'shape' of the curve computed is indeed largely similar. Furthermore, cases of word occurrences in a local context having distinct word

Figure 2: Comparison of de_{3gr} and en_{3gr}



senses appear to be rare in practice (Gale et al., 1992a).

4.1 Measuring similarity

For each language, a vector of similarity measurements of paragraphs is generated using the method described in section 2.2. The measurements of the three versions of the UBS document can be viewed as three approximations of a single, 'underlying' discourse structure. The similarity of the three vectors of measurements is shown graphically by plotting the measurements.

To actually quantify the similarity of the paragraph similarity measurements, we computed the correlation of two arrays of measurements using a discrete summing technique. This results in an array h of correlation coefficients:

$$h_j = \sum_{k=0}^{n_h-1} x_{(j+k)} y_k$$

for $j = 0, 1, 2, \ldots, n_h - 1$ and $n_h = n_x + n_y - 1$. Here, n_h is the total number of points to be output from the correlation routine, and n_x ($= n_y$ in our case) the

number of points in the x array (DXML, 1993). The values in the h array can be normalized to fit in the interval $[0, 1]$ by dividing the values by the product of the norm of the x and y vectors. Only the first correlation coefficient is relevant, because there are no phase shifts, since the x and y input arrays are perfectly parallel.

Variation of analysis methods (e.g. whether or not words are lemmatized) yields slightly different paragraph similarity vectors. The correlation routine can be used to quantify the effect of using another analysis method on discourse structure, and to determine how similar various language versions of a single document are, in terms of discourse similarity. One can also turn the argument around and use correlation as a guideline to evaluate different analysis methods. If morphological analysis is hypothesized to help detect lexical cohesion, then two language versions of a document should be more strongly correlated when index terms are selected using morphological analysis.

The correlation of the three arrays of paragraph similarity measurements, after lowpass filtering, is shown in a correlation matrix. In the matrix shown in figure 3 we have included two analyses of German, one with (de_m), and another without (de_{nm}) lemmatization, two analyses of the English, and two analyses of French, one of which is based of trigram morphology (fr_{3g}).

As shown, morphological analysis, and even character trigram analysis[4], resulted in a consistent, but small, improvement in measuring the similarity of

[4]The trigram analysis in turn improved on an analysis based on character fourgrams, fivegrams, and sixgrams (in that order).

Figure 3: Correlation of trilingual corpus

	de_{nm}	de_m	en_{nm}	en_m	fr_{nm}	fr_{3g}
de_{nm}	1					
de_m	0.976	1				
en_{nm}	0.86	0.90	1			
en_m	0.87	0.90	0.996	1		
fr_{nm}	0.80	0.81	0.87	0.88	1	
fr_{3g}	0.80	0.82	0.91	0.92	0.95	1

	de_{3gr}	en_{3gr}	fr_{3gr}
de_{3gr}	1		
en_{3gr}	0.97	1	
fr_{3gr}	0.94	0.94	1

the documents. The ngram analysis turned out to be superior to the analysis derived by lemmatization of unambiguous inflected word forms (cf. figure 3). Apparently, the lemmatization technique used misses a considerable number of morphological relations that can be captured with ngram analysis.

It will now be clear why the paragraph alignment phase was applied to the corpus before further analysis. Without alignment, the measurements would not be comparable, and the correlation measure would be meaningless.

4.2 Parallel Segmentation

Instead of computing the correlation of the representations as a measure of document similarity, it is also possible to use the representations for text segmentation as in TextTiling. Minimal values are detected and used as segment boundaries. One can then check whether representations of distinct language versions have a similar segment structure, i.e. if there is a transition from one

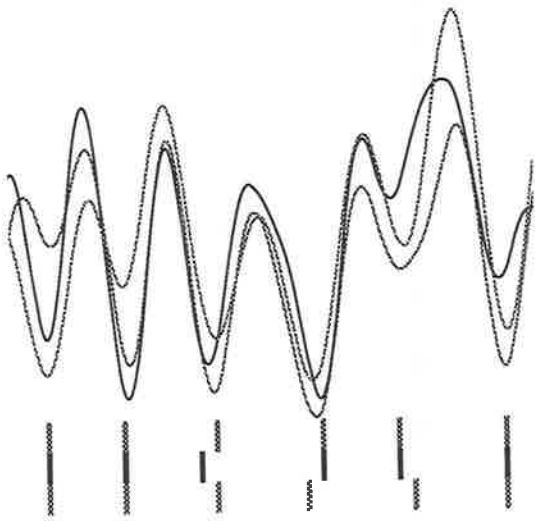
subtopic to another in the discussion, then this transition should be detected in all three documents.

Obviously, this is a weaker notion of similarity than discrete correlation because much information in the representations is ignored, because the paragraph similarity measurements are replaced by boolean values, viz. whether or not a gap between two paragraphs is a sub-topic boundary. Furthermore, agreement should be normalized for segment length, because distortions in segmentation are more likely to occur when segments are longer. This is the case when the representations are modified by lowpass filtering.

Parallel segmentation can be implemented in various ways. The reliability or 'strength' of a boundary can be determined by checking whether the boundary is confirmed by other language versions. If the measurements on three documents indicate a segment boundary between two paragraphs, then one will be fairly confident that there is indeed a transition to another subtopic, whereas if only one document indicates a boundary, then this is probably an incorrect measurement. We also found a number of cases of weak distortions, where two languages agree on a boundary, and the third one puts the boundary one paragraph earlier or later. An example of this is given in figure 4, where we have indicated some occurrences of segment boundaries in three language versions. These are near misses that a segmentation tool could detect and correct automatically.

Although clear correspondence between documents is an indication that the texts are related by translation, lack of correspondence (overall, or locally) can

Figure 4: Weak distortions (segment boundaries for de_{3gr}, en_{3gr} and fr_{3gr})



result from various reasons. One reason could be an alignment error, or a serious translation error (e.g. an untranslated section), but translation to synonyms, ambiguity, and errors in morphological analysis could cause distortions locally even when the translation is basically correct. In the specific case of the UBS corpus, some local distortions arise when there really is hardly any multi-paragraph structure at all, such as when a sequence of paragraphs is an enumeration of short overviews of economic developments in widely different sectors in the economy.

5 Discussion

Obviously, the quantitative techniques we have used in this paper to perform comparative discourse analysis of parallel texts are very unsophisticated, using only a subset of lexical cohesion relations, and ignoring all sub-paragraph structure. We have not evaluated the method with text units smaller than paragraphs.

Although some improvements can be obtained, esp. by applying a wider range of lexical cohesion relations, the parallel discourse analysis method discussed in this paper is best viewed as a pre-processor for other tasks. Some areas in which it can be applied are text translation alignment, translation studies, evaluation of subtopic structuring techniques, and (monolingual and multilingual) tools that use distributional information from text corpora.

To date, systems exist that align translated documents at several levels of granularity. The first papers focussed on alignment at the sentence level (Brown et al., 1991). Recent papers have discussed alignment and correspondences at the level of words (Dagan et al., 1993) or phrases (van der Eijk, 1993). This paper has discussed how these techniques could be complemented with an algorithm to align texts at a multi-paragraph level. Earlier alignment algorithms have been remarkably successful in using only the length of parallel segments. Other algorithms have also taken lexical distribution into account (Kay and Roescheisen, 1993; Chen, 1993). Since the discourse similarity measurements have been shown to correlate strongly, this measure could also be used by align-

ment algorithms that can measure text-translation similarity based on more than one parameter, thus combining evidence based on segment size, lexical information, and discourse cohesion similarity. The computational overhead needed to compute cohesion similarity is very limited.

Subtopic structuring of large documents is useful for a variety of applications. Effectiveness of information retrieval on full-length documents has been shown to improve by taking advantage of document structure (Hearst and Plaunt, 1993). The availability of parallel corpora (where discourse structure is preserved in the translation) will greatly help designing and evaluating such text segmentation systems.

Disambiguation algorithms such as (Yarowsky, 1992) that train on arbitrary-size text windows and algorithms that use lexical co-occurrence to determine semantic relatedness (Schuetze and Pedersen, 1993) might also benefit from using windows with less arbitrary boundaries. This naturally extends to similar algorithms that use distributional information in parallel corpora (Gale et al., 1992b).

References

Brown, P., Lai, J., and Mercer, R. (1991). Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 169–176.

Chen, S. (1993). Aligning sentences in bilingual corpora using lexical infor-

mation. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 9–16.

Dagan, I., Church, K., and Gale, W. (1993). Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pages 1–8.

DXML (1993). *Digital Extended Math Library for DEC OSF/1 AXP*. Digital Equipment Corporation, Maynard, Massachusetts.

Gale, W., Church, K., and Yarowsky, D. (1992a). One sense per discourse. In *Proceedings of the Darpa Speech and Natural Language Workshop*, pages 233–237.

Gale, W., Church, K., and Yarowsky, D. (1992b). Using bilingual materials to develop word sense disambiguation methods. In *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 101–112, Montréal.

Hearst, M. (1993). TextTiling: a quantitative approach to discourse segmentation. Technical Report 93/24, Project Sequoia, University of California, Berkeley.

Hearst, M. and Plaunt, C. (1993). Subtopic structuring for full-length document access. In *Proceedings of SIGIR*.

Kay, M. and Roescheisen, M. (1993). Text-translation alignment. *Computational Linguistics*, 19(1):121–142.

Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Salton, G. and McGill, M. (1983). *Introduction to Modern Structured Information Retrieval.* McGraw-Hill.

Schuetze, H. and Pedersen, J. (1993). A vector model for syntagmatic and paradigmatic relatedness. In *Proceedings of the Ninth Annual Conference of the UW Centre for the New OED and Text Research*, pages 104–113.

van der Eijk, P. (1993). Automating the acquisition of bilingual terminology. In *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–119.

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING)*, pages 454–460.