Example-Based Machine Translation: An Investigation

STEVEN S NGAI · RANDY B GULLETT

CS224N FINAL PROJECT

Problem

Now more than ever, the world looks to computers to perform the task of translation. Spurred on by the information age, more and more computer-enabled sources are pouring an increasing proportion of documents into global forums including, but not limited to, the Internet. Forums like these are becoming sources of information for a growing number of people worldwide, and it is no surprise that everyone wants his information in his own language. Often, restrictions on the accuracy of these translations have become tighter: bodies like the European Union produce daily proceedings that, by law, must be translated into all languages of its constituent countries so precisely that any translation can be used in a court of law. Not surprisingly, human translators are unable to keep up with this demand.

Among machine translation systems, traditional transformational methods are somewhat difficult to contruct, as they basically involve hardcoding the idiosyncrasies of both languages. But through the work of human translators, large parallel corpora have become available. Therefore it makes sense, if it is viable, to base translations off these large bodies of text—this in order somehow to capture the knowledge contained in preexisting translations. Our investigation attempts to look into one such method and its successes and failings.

A Proposed Solution

Example based machine translation (EBMT) is one such response against traditional models of translation. Like Statistical MT, it relies on large corpora and tries somewhat to reject traditional linguistic notions (although this does not restrict them entirely from using the said notions to improve their output). EBMT systems are attractive in that they require a minimum of prior knowledge and are therefore quickly adaptible to many language pairs.

The particular EBMT system that we are examining works in the following way. Given an extensive corpus of aligned source-language and target-language sentences, and a source-language sentence to translate:

1. it identifies exact substrings of the sentence to be translated within the source-language corpus, thereby returning a series of source-language sentences
2. it takes the corresponding sentences in the target-language corpus as the translations of the source-language corpus (this should be the case!)
3. Then for each pair of sentences:
4. it attempts to align the source- and target-language sentences;
5. it retrieves the portion of the target-language sentence marked as aligned with the corpus source-language sentence's substring and returns it as the translation of the input source-language chunk.

The above system is a specialization of generalized EBMT systems. Other specific systems may operate on parse trees or only on entire sentences.

The system requires the following:

1. Sentence-aligned source and target corpora.
2. Source- to target- dictionary
3. (Stemmer)

The stemmer is necessary because we will typically find only uninflected forms in dictionaries. While it is consulted in the alignment algorithm, it is not consulted in the matching step—as stated before, those matches must be exact.

In this project we rely on papers published by Ralf D. Brown and by Sergei Niremburg describing work on the PanGloss translation project. Their two approaches are different, but nevertheless provided a good guideline for our implementation.

Methods (Algorithms)

Indexing

In order to facilitate the search for sentence substrings, we need to create an inverted index into the source-language corpus. To do this we loop through all the words of the corpus, adding the current location (as defined by sentence index in corpus and word index in sentence) into a hashtable keyed by the appropriate word. In order to save time in future runs we save this to an index file.

Chunk searching and subsuming

```
Keep two lists of chunks: current and completed.
Looping through all words in the target sentence:
        See whether locations for the current word extend any chunks on the current list
        If they do, extend the chunk.
        Throw away any chunks that are 1-word. These are rejected.
        Move to the completed list those chunks that were unable to continue
        Start a new current chunk for each location
At the end, dump everything into completed.
```

```
Then, to prune, run every chunk against every other:
        If a chunk properly subsumes another, remove the smaller one
        If two chunks are equal and we have too many of them, remove one
```

Alignment

The alignment algorithm proceeds as follows:

1. Stem the words of specified source sentence
2. Look up those words in a translation dictionary
3. Stem the words of the specified target sentence
4. Try to match the target words with the source words—wherever they match, mark the correspondence table.
5. Prune the table to remove unlikely word correspondences.

6. Take only as much target text as is necessary in order to cover all the remaining (unpruned) correspondences for the source language chunk.

Stemming is done using .
RANDY YOUR STUFF GOES HERE.
Pruning is done using .

The pruning algorithm relies on the fact that *single* words are not often violently displaced from their original position. This assumption is true between English and most of the Romance languages; however, notable exceptions may (but not necessarily) include the oft-cited non-SVO languages Korean, Japanese, and Arabic. In addition, the pruning algorithm works best when most word correspondences are 1-to-1.

Implementation
The project is implemented in Java.
The corpus was prepared using a small Perl script and command-line tools; it was finalized by hand.

Corpus
We used English-Spanish texts from the Pan American Health Organization as our bilingual corpus. To select files for this purpose, we examined the files and chose those which seemed to be reports, summaries, or speeches. These types of documents have large amounts of running text; therefore we judged them most likely to align with minimal human assistance.

We avoided files heavy in charts or in list formatting, such as resolutions. Perhaps these documents, by way of their specificity and precision of wording, may have produced more literal translations. However, we would want to reliably identify section markers, use the items and sections as alignment anchors, yet remove them afterwards, a task that might be interesting to investigate as a automated processing task but one which we did not have the time to implement.

Difficulties in alignment
We used the following sequence of command-line text-processing commands to preprocess both Spanish and English:

```
tr '\n' '@'< $spanishfile | sed 's/Dr./Dr/g' | sed
's/Mr./Mr/g' | sed 's/"¿//g' | sed 's/--/~/g' | sed
's/@@[@]*/=/g' | sed 's/-@//g' | sed 's/@/ /g' | sed 's/=/
/g' | sed 's/\.=/\./g' | tr ':;.~?' '\n' | sed 's/, / , /g'
| sed 's/(/( /g' | sed 's/)/ )/g' > $outspanishfile
```

This transformed the source text into a one-sentence-per-line document with varying amounts of whitespace. The substitutions produce a few intermediate symbols to simplify things. We broke "sentences" on colons and em dashes too, figuring that they would provide valuable anchor points for sentences; we also spaced out commas and parentheses.

Manual processing involved spotting the beginnings of each labeled "sentence" to ensure that they lined up. If they did not, we would delete or (preferentially) break the longer line to match the shorter, since most misalignments were of type 1 and 3. In the end the Spanish and English corpus files have the exact same line count.

Naturally, not all translations are literal, and therefore we expected the task to be fairly difficult. However, it seemed as if the translators did a rather straight translation for a majority of the reports. For instance, Spanish Document 0119, a report, required absolutely no editing whatsoever and remained in the corpus in its original form.

We encountered mainly the following types of misalignments:

1. Differing pronunciation (; vs ,) to separate lists, which caused one sentence to break and not the other;
2. Rephrasings of several sentences;
3. Fragmenting of Spanish sentences into what were technically sentence examples in English, ie. *For this is true.* Spanish tends to permit longer sentences.

The translators took most liberties in translating speeches, such as Document 0002. We suspect this was to preserve their dramatic and rhetorical force. We found many examples of all misalignments of types 1, 2, and 3 above.

At the same time there were some complete surprises too. Following is an example of a footnote that, without warning, appeared in the middle of Spanish text to explain the acronym OPS:

```
que hayan afectado a nuestra región, el sector salud y los
trabajadores de salud del Perú supieron responder
afirmativa y exitosamente, creando un cuerpo de
experiencias y de

__________________
*Organización Panamericana de la Salud, Oficina Regional
para las Américas de la Organización Mundial de la
Salud

conocimientos que ha servido para que los demás países de
la Región de las Américas, afectados después del Perú po
```

We also had to check manually for translator's notes at the beginnings and ends of documents:

```
[TRANSLATOR'S NOTE -  See Article 11 re settlement of dis-
putes by arbitration.  The Spanish text obviously considers
three (3) parties to this Agreement; however, there are
only
two (2).  The Translation to English correctly states the
number of parties and of arbitrators - it is recommended
that
the Spanish text be corrected.]

   [TRANSLATION OF DOCUMENT   EOO97.FIN]
```

Files Included

Chunk.java	represents a matching chunk of a source-language corpus sentence
ChunkFinder.java	object that searches the corpus for matching chunks
ChunkPruner.java	object that prunes subsumed chunks
FileUtils.java	contains a few functions for writing the index to file
IndexedWord.java	represents a source word and its index data into the corpus
Indexer.java	object that takes the corpus and forms the index
Word.java	represents a word once its corpus sentence source has been fixed
Process.pl	the pre-processing Perl script
*.eng	English corpus files
*.span	Spanish corpus files
*.index	index files for corpora

In the parser directory are files that successfully implement a Earley chart parser. We had developed these files expressly for the purpose of this project, but when our direction changed we were unable to use them:

Grammar.java	manages the grammar of the parser
Parser.java	object that coordinates the top-level activities of the Earley parser
State.java	represents a parse state
Tag.java	represents a Tag.
Chart.java	represents one of the n charts in a parse of a sentence of length n
CategoryTag.java	represents a category. Subclasses Tag.
WordTag.java	represents a word. Subclasses Tag.
POSTag.java	represents a POS tag. Subclasses Tag.
Rule.java	represents a grammar rule.
*.lexicon, *.grammar	files to load a grammar

This parser is set up to demonstrate the parse of a sentence from the last homework.
Type: `java Parser p3.lexicon p3.grammar`

Linguistic models and their validity

EBMT relies on the assumption that large matching chunks of text give enough clues about the context to correctly translate a sentence (or at least a chunk). For instance, in a translation from English to Spanish, an English verb alone is not enough to determine what the verb inflection is in Spanish, but once we expand the English chunk to include the subject, then we know the person to which it should be inflected. Therefore if two chunks both contain this same information, we can expect their proper translations to be the same as well. Or we may not know what sense a word is used in, but as soon as we obtain a few words surrounding the word in question, we can figure out whether, for instance, we are measuring in feet or walking on them. By such context clues, EBMT systems can overcome problems of word sense disambiguation, agreement, and even idiom.

Of course there is the danger of a spurious match, e.g. *Dogs bite* for *Let the children who hate dogs bite them back*, which would give a very different translation for the verb. However, one hopes that by procuring large corpora, we will produce better, longer matches. And with longer matches, there is a decline in the probability that a sequence of words will happen to occur in a different grammatical relation.

Therefore the model employed is most similar to, and as valid as, the n-gram model.

Design decisions

Their Justification

1. Allow chunk matching over sentence boundaries? **No.**

Indeed, the large corpus argument holds, and technically the end of one sentence has some role in linking to the beginning of another, but it is not as strong as within a sentence. Primarily this is because a sentence is held together by semantics as well as syntax; between sentences, and in discourse, only the semantics remains. Furthermore, allowing chunk matching over boundaries would greatly complicate the process both of indexing and aligning chunks.

2. Index punctuation? **Yes**.

Intra-sentence punctuation (primarily commas and parentheses) should theoretically help the alignment algorithm to match the sentence with its counterpart. The only danger is that things could go significantly wrong if the intra-sentence punctuation does not match. Indeed, there seem to be sentences where this is true.

3. Create a list of stop words? **No**.

Niremburg's implementation of the system uses a list of stop words. As one manifestation of this he does not index the stop words, but because of that he can no longer track whether corpus chunks are truly continuous or separated by an arbitary number of such words. Since the principle of a close match plays such a large part in this type of MT, we have chosen to go with the surer method, even if it does require more resources.

4. Manual correction of alignment? **Yes**.

We tried our best not to improve unduly the quality of our translation, but to have egregious misalignments wouldn't really help us to produce a coherent report on the strengths of such a MT system.

5. Equivalence classes? **No**.

Brown describes the use of equivalence classes like PERSON, DATE, and PLACE. But lists of these—particularly of important PERSONs—are unlikely to be worth the effort spent in compiling them.

Testing

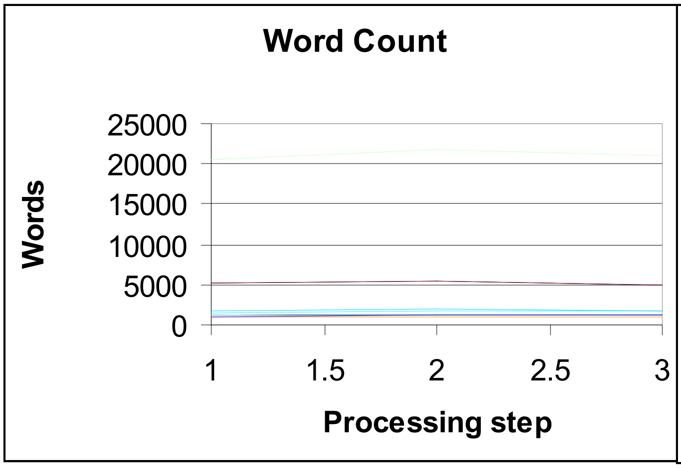
The primary testing for this system consisted of attempting to translate sentences from another document randomly chosen from those judged to be suitable. We judged the linguistic correctness of the returned translation, determined the percentage cover of the sentence, and analysed the types of mistakes that the system made.

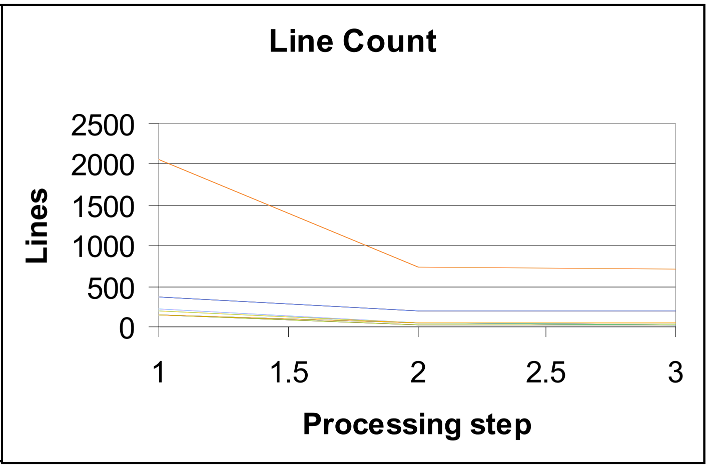
Results

Effect of Corpus Preparation

Word counts of some sample documents.

FINAL	PREPROCESSED	ORIGINAL
197 5062 30053 0002.eng	203 5434 32237 0002.eng	368 5111 32005 e0002.eng.tr
36 1104 7267 0117.eng	39 1111 7369 0117.eng	140 1025 7301 e0117.eng.tr
34 1085 7038 0118.eng	35 1092 7128 0118.eng	139 1014 7073 e0118.eng.tr
50 1655 10851 0119.eng	51 1704 11230 0119.eng	209 1582 11133 e0119.eng.tr
40 1310 8518 0120.eng	40 1312 8525 0120.eng	159 1218 8451 e0120.eng.tr
197 5063 30335 0002.span	204 5439 32565 0002.span	379 5128 32351 e0002.dos.tr
36 1214 7669 0117.span	38 1219 7835 0117.span	147 1144 7774 e0117.dos.tr
34 1172 7564 0118.span	33 1176 7645 0118.span	142 1106 7594 e0118.dos.tr
50 1843 11393 0119.span	50 1891 11753 0119.span	208 1784 11663 e0119.dos.tr
40 1423 8786 0120.span	40 1423 8786 0120.span	156 1345 8724 e0120.dos.tr
714 20931 129474 total	733 21801 135073 total	1015 9950 65963 totalE
		1032 10507 68106 totalS
		2047 20457 134069 total





The count of words goes up from the original to the preprocessed because of separation of punctuation. Like all the other statistics, it goes down going into the final corpus because of extraneous line removal.

Performance of the Indexer

```
date ; java Indexer test1.eng test1.eng.index ; date
Fri Jun  7 02:02:56 PDT 2002
--Done!
```

```
      Fri Jun  7 02:02:57 PDT 2002
```

We processed this sample index of 10000 words in less than a second. To extend the corpus to normal corpus sizes (millions of words) should not take unreasonably long.

The size blowup for the preceding is as follows:

```
      63727 test1.eng
      86762 test1.eng.index
      65747 test1.span
      94325 test1.span.index
```

In each case the size of each file increases by a factor of .4. Because we save on hashtable lookup time when we come back and load up the index—we only hash once—loading the index once instead of recomputing the index will make it worth using, especially as corpus size increases.

Performance of the ChunkFinder

Remember that many returned chunks have been subsumed, and also that two or more consecutive words must match in order to be a chunk. **Bold** text indicates text that was matched in some substring or another.

Sentence 1: (28/42 words matched = 67% coverage, avg len = 22/7 = 3.14 words)
The need to optimize excessive health expenditures to solve problems **of public health and social** orientation **continued to be a priority for the Governments of the** Netherlands Antilles **, an** autonomous **part of the** Kingdom **of the** Netherlands **, and of** Aruba

```
the:(0) (202,0) need:(1) (202,1)
the:(0) (331,14) need:(1) (331,15)
the:(0) (355,6) need:(1) (355,7)
need:(1) (40,11) to:(2) (40,12)
need:(1) (130,3) to:(2) (130,4)
need:(1) (139,4) to:(2) (139,5)
of:(10) (198,21) public:(11) (198,22) health:(12) (198,23) and:(13) (198,24) social:(14)
(198,25)
of:(10) (227,9) public:(11) (227,10) health:(12) (227,11) and:(13) (227,12) social:(14)
(227,13)
of:(10) (282,18) public:(11) (282,19) health:(12) (282,20) and:(13) (282,21) social:(14)
(282,22)
continued:(16) (344,34) to:(17) (344,35) be:(18) (344,36)
to:(17) (302,5) be:(18) (302,6) a:(19) (302,7) priority:(20) (302,8) for:(21) (302,9)
the:(22) (302,10)
the:(22) (12,43) governments:(23) (12,44)
the:(22) (24,55) governments:(23) (24,56)
the:(22) (70,9) governments:(23) (70,10)
of:(24) (1,7) the:(25) (1,8)
,:(28) (21,1) an:(29) (21,2)
,:(28) (136,49) an:(29) (136,50)
,:(28) (234,6) an:(29) (234,7)
```

```
part:(31) (57,4) of:(32) (57,5) the:(33) (57,6)
part:(31) (82,18) of:(32) (82,19) the:(33) (82,20)
part:(31) (88,13) of:(32) (88,14) the:(33) (88,15)
part:(31) (187,11) of:(32) (187,12) the:(33) (187,13)
part:(31) (297,34) of:(32) (297,35) the:(33) (297,36)
of:(35) (1,7) the:(36) (1,8)
,:(38) (272,15) and:(39) (272,16) of:(40) (272,17)
```

Sentence 2: (22/34 = 65% coverage, 23/9 = 2.55 words)
PAHO/WHO collaborated with the authorities in developing and strengthening local health systems , in executing specific programs for vulnerable populations **, and in** increasing **primary care** activities through community **organization to** solve local **problems**

```
paho/who:(0) (261,10) collaborated:(1) (261,11) with:(2) (261,12) the:(3) (261,13)
paho/who:(0) (333,0) collaborated:(1) (333,1) with:(2) (333,2) the:(3) (333,3)
authorities:(4) (200,18) in:(5) (200,19)
authorities:(4) (201,5) in:(5) (201,6)
authorities:(4) (333,5) in:(5) (333,6)
in:(5) (259,6) developing:(6) (259,7)
in:(5) (281,17) developing:(6) (281,18)
in:(5) (346,4) developing:(6) (346,5)
and:(7) (226,22) strengthening:(8) (226,23)
and:(7) (301,22) strengthening:(8) (301,23)
and:(7) (304,50) strengthening:(8) (304,51)
local:(9) (336,32) health:(10) (336,33) systems:(11) (336,34) ,:(12) (336,35)
,:(12) (8,40) in:(13) (8,41)
,:(12) (342,2) in:(13) (342,3)
,:(12) (342,13) in:(13) (342,14)
,:(20) (177,24) and:(21) (177,25) in:(22) (177,26)
,:(20) (213,19) and:(21) (213,20) in:(22) (213,21)
,:(20) (254,27) and:(21) (254,28) in:(22) (254,29)
,:(20) (287,24) and:(21) (287,25) in:(22) (287,26)
primary:(24) (250,31) care:(25) (250,32)
primary:(24) (320,14) care:(25) (320,15)
organization:(29) (262,7) to:(30) (262,8)
```

Sentence 3: (13/24=54% coverage, 15/7=2.14 words)
Several workshops on **community participation were** held **, and this strategy was** applied **in the programs to prevent** drug abuse and alcoholism in CuraÛao

```
community:(3) (251,77) participation:(4) (251,78)
community:(3) (292,2) participation:(4) (292,3)
participation:(4) (275,61) were:(5) (275,62)
,:(7) (84,38) and:(8) (84,39) this:(9) (84,40)
,:(7) (144,10) and:(8) (144,11) this:(9) (144,12)
```

```
strategy:(10) (278,1) was:(11) (278,2)
in:(13) (4,14) the:(14) (4,15)
in:(13) (353,3) the:(14) (353,4)
in:(13) (354,7) the:(14) (354,8)
the:(14) (229,3) programs:(15) (229,4)
to:(16) (261,21) prevent:(17) (261,22)
to:(16) (311,0) prevent:(17) (311,1)
to:(16) (355,12) prevent:(17) (355,13)
```

Sentence 4: (13/31 = 42%, 14/5=2.8words)
As a result of this experience **, PAHO/WHO** sponsored a workshop in St. Martin attended by members **of the community** in that island **as well as** from St. Eustatius and Saba

```
as:(0) (203,0) a:(1) (203,1) result:(2) (203,2) of:(3) (203,3)
as:(0) (237,17) a:(1) (237,18) result:(2) (237,19) of:(3) (237,20)
as:(0) (241,0) a:(1) (241,1) result:(2) (241,2) of:(3) (241,3)
as:(0) (325,0) a:(1) (325,1) result:(2) (325,2) of:(3) (325,3)
of:(3) (2,35) this:(4) (2,36)
of:(3) (315,17) this:(4) (315,18)
of:(3) (356,7) this:(4) (356,8)
,:(6) (259,3) paho/who:(7) (259,4)
,:(6) (337,3) paho/who:(7) (337,4)
,:(6) (345,5) paho/who:(7) (345,6)
of:(17) (319,43) the:(18) (319,44) community:(19) (319,45)
as:(23) (200,13) well:(24) (200,14) as:(25) (200,15)
as:(23) (300,39) well:(24) (300,40) as:(25) (300,41)
as:(23) (328,23) well:(24) (328,24) as:(25) (328,25)
as:(23) (348,15) well:(24) (348,16) as:(25) (348,17)
```

Sentence 5: (10/25 = 40% coverage, 11/5=2.2 words)
These **and other** activities helped increasingly bring to light **the need for establishing** greater collaboration **among the** six islands and mutual support **in health** matters .

```
and:(1) (128,14) other:(2) (128,15)
and:(1) (285,33) other:(2) (285,34)
and:(1) (299,29) other:(2) (299,30)
the:(9) (202,0) need:(10) (202,1) for:(11) (202,2)
the:(9) (331,14) need:(10) (331,15) for:(11) (331,16)
the:(9) (355,6) need:(10) (355,7) for:(11) (355,8)
for:(11) (234,19) establishing:(12) (234,20)
for:(11) (280,4) establishing:(12) (280,5)
among:(15) (308,6) the:(16) (308,7)
in:(22) (169,11) health:(23) (169,12)
in:(22) (347,34) health:(23) (347,35)
in:(22) (351,18) health:(23) (351,19)
```

We see clearly outlined the poverty of our training. While there are some useful field-specific chunks isolated, much of what is returned consists of little functional words. True, we have about 50% coverage, but as we know each extra percentage point becomes harder to gain. The average length of chunk is around 2.3.

RANDY YOUR STUFF GOES HERE

Failures and Reasons

The performance of the aligner was hampered by a non-ideal dictionary (we are not sure why, but what kind of dictionary doesn't list *de* as a translation of *of*?). The effect of the non-ideal dictionary was especially prominent when we removed the code that attempts to guess at missing words (because the dictionary was too poor, and too many guesses were being made).

The performance of the chunk finder was hampered by an inadequate corpus. In this case, though, it would have been very time-consuming to check the alignment of sentences.

Suggestions for Improvement

It would be interesting to determine the added utility of supplying word-for-word translations for the remaining words.

A very apparent failing of this system is that there is no way to combine chunks at the end. (What readability currently exists does so only because Spanish and English roughly share a word order.) A possible next step might be to note the syntactic class(es) represented by the words in question. Using transformational techniques, we could then attempt to reconstruct the sentence properly. The problem is that our chunks may represent random pieces of trees, e.g. saw the man who often, making it difficult to use any tree paradigm with them. The other option is to require that chunks be well-formed pieces of trees, but that requirement reduces the wide-ranging utility of the system.

Responsibilities

Steve – Indexing half, Corpus alignment (preprocessing), Manual Postprocessing, Parser
Randy – Alignment half, Integration, Manual Postprocessing

References

1. Brown, Ralf D. Example-Based Machine Translation in the Pangloss System. Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING-96)

2. Nirenburg, Sergei et al. A Full-Text Experiment in Example-Based Machine Translation Proceedings of the International Conference on New Methods in Language Processing, Manchester, England, pp. 78-87. (1994)