PECAT: A COMPUTER-AIDED TRANSLATION TOOL
BASED ON BILINGUAL CORPORA

LE SUN, YIBO ZHANG, JUNLIN ZHANG, YUFANG SUN

Chinese Information Processing Center, Institute of Software
Chinese Academy of Sciences, Beijing 100080, P. R. China
lesun, zyb, zjl, yfsun@sonata.iscas.ac.cn

Abstract

With the widespread use of computer in translation work and daily life, the bilingual
corpora become more and more. In this paper the PECAT (Pilot English-Chinese Computer-Aided
Translation) system based on bilingual corpora is described. There are mainly three modules in our
system: corpus-processing module, matching module and post-edit module. In order to increase
the coverage to input source sentence, the text
alignment in corpus-processing module is based
on chunk level. The matching algorithm of input
sentence and source sentence is a three-layer Edit
Distance algorithm guided by user, which include
the information of word morph and word POS.
Preliminary experiments show promising results.

Keywords:
Computer-Aided Translation, Bilingual Corpora,
Text Alignment, Matching Algorithm

1 Introduction

With the widespread use of computer in translation work and daily life, the bilingual
corpora become more and more. At present, there
are many research works about how to take
advantage of the bilingual corpora and use them
to machine translation system or computer-aided
system as a knowledge base.

Since Kay’s paper 1], various translation tool
or TWB (Translation Workbench) have been
proposed. The CAT systems, such as Trados
Translation Workbench, IBM Translation
Manager, Star Transit and Eurolang Optimizer,
Atril DejaVu, SDL SDLX, are prevalence among
translators. However, these commercial CAT
tools do not answer the question about the proper
place of man and machines in language
translation completely. As Macklovitch 18 has
pointed out "... these system can exploit only a
small part of the translation knowledge lying
dormant in past translations." Are there more
things we can do for translators? With the
development of EBMT, SBMT and alignment
technology in bilingual corpora, the answer is
sure.

There are two CAT tools specially designed
for translation between English and Chinese at
the least. One is a commercial CAT system,
named Ya Xin from Ya Xin Cheng soft company.
The other is the experimental CAT system,
named PENS 19 from Microsoft Research China.
The former is very simple and use TM
(Translation Memory), in which the stored
translation unit is sentence, as its database. There
are several large English-Chinese bilingual
technical dictionary in it. The latter is more
powerful and used for Chinese users to write
English. The word spelling help by statistical
approach and suggestive example sentences
recommendation are two notable functions.

In this paper we propose the PECAT (Pilot
English-Chinese Computer-Aided Translation)
system based on bilingual corpora, used by
technical document translators, such as the
freelance translators in localization company.
There are mainly three modules in our system:
corpus-processing module, matching module and
post-edit module. The most important factor we
care in corpus-processing module is the precise of
corpus processing, however in matching module
it's the speed to search similar sentence, and in
post-edit module it's the benefit of the
recommended information by system. In order to
increase the coverage to input source sentence,
the text alignment in corpus-processing module is based on chunk level. The matching algorithm of input sentence and source sentence is a three-layer Edit Distance algorithm guided by user, which include the information of word morph and word POS.

In the following sections, we fist give an overview of the system, introduce the main components: corpus-processing module, matching module and post-edit module. Then in section 3 describe the alignment principles we used for chunk alignment. In section 4 the user-guided three-layer matching algorithm of input sentence and source sentence is presented in detail. Then the experiment results of our pilot system are showed with the error analysis. Finally we concluded our paper with future works.

2 System Overview

![Diagram of System Architecture of PECAT]

As show in figure 1, the PECAT system is composed by three modules: corpus-processing module, matching module and post-edit module. In corpus-processing module the English text and corresponding Chinese text will be aligned form paragraph level to sentence level and then to chunk level. Before the alignment processing, the English texts have been tagged with POS tag, and the Chinese texts have been segmented and tagged with POS tag. The corpus processing precise is stressed in this module. All the chunk aligned bilingual corpora are stored in translation memory with convenient index.

While input a source sentence, a program will do morph analysis and add POS tag to this sentence and try to separate it into a few chunks. Then the multi-layer matching program will retrieve the most similar sentence or chunks in translation memory within a second. The retrieval speed is the key-point we consider in this module.

Then generate the reference sentence to user within a convenient edit environment. The user can accept or refuse or revise the reference sentence or chunks the system recommends. After manual revise, output the target sentence and combine with the input source sentence together to bilingual corpora automatically. In later sections the alignment principle used in corpus-processing module and user-guided matching algorithm in matching module will be presented in detail.

3 Alignment Principle

Different NLP applications need different bilingual corpora, which are aligned at different level. They can be divided by the nature of the segment to section level, paragraph level, sentence level, phrase or chunk level, word level, byte level, etc.

Many researchers have studied the text alignment problem recently and a number of quite encouraging results have been reported to different level alignments.

There are basically three kinds of approaches on text alignment: the statistical-based approaches \([4,5]\), the lexicon-based approaches \([6,7]\), and the combine of them \([8,9]\).

As for our application, namely a computer-aided translation tool for technical document translators, we choose the combination of the statistical-based approach and the lexicon-based approach for its robust and better performance. The alignment level we select is chunk level for following reasons:

1) To increase the input source sentence coverage. It does seldom find exactly same or nearly same sentence during translating except the manual update. So it's needed to align the bilingual corpora below the sentence level, on which we can acquire term or phrase and their useful translation information.
2) The word level alignment between English and Chinese language is difficult to deal with. There are no cognate words. The change in Chinese word order and word POS always produce many null and mistake correspondences.

3) It's a common phenomenon, especially in technical document, that when we translate the English sentence to Chinese sentence, all the words in one English chunk tend to be translated as one block of Chinese words that are coterminous. There are stronger boundaries between chunks than between words when we translate texts.

4) As we all known, chunk can be assigned syntactic structure, which comprises a connected sub-graph of the sentence's parse tree. So it's possible to align sentence structure and obtain translation grammars based on chunks by parsing in future.

In table 1 the outline of alignment algorithm is showed.

---

Step 1: Align the English and Chinese bilingual document in paragraph level by use of bilingual lexicon with heuristic algorithm.

Step 2: Use the length-based algorithm and bilingual lexicon together to align sentence within paragraph automatically and then the results are corrected by human with the help of system.

Step 3: According to the definition of Chunk in English, separate the English sentence into a few chunks and labeled with order number from left to right.

Step 4: Try to find the Chinese translation of every English chunk created in step 3 by bilingual dictionary and synonymy Chinese dictionary. If the Chinese translation is find, then label the Chinese words with the same number used for the English chunk in step 3.

Step 5: Disambiguate the multi-label Chinese words by the translation location of coterminous words within the same English chunk.

Step 6: Separate the Chinese sentence into a few chunks by heuristic information.

Step 7: Save the English chunk and corresponding Chinese chunk together as a translation unit in translation memory.

---

Table 1 Outline of Alignment Algorithm

We first align the English and Chinese bilingual document in paragraph level by a heuristic algorithm. Then we use the length-based algorithm and bilingual lexicon together to align sentence within paragraph automatically and the results are corrected by human with the help of system. Then identify the chunks of English sentences by shallow parser and predict the chunk boundaries of Chinese sentences from the translation of every English chunks and heuristic information by use of the bilingual lexicon. The ambiguities of Chinese chunk boundaries are resolved by the coterminous words in English chunks. For detail about Chunk alignment, please see paper [12].

In this pilot system we separate the English sentence into a few chunks base on following considerations:

1) The grammar used for English shallow parser.

2) The more times the several words appeared together in corpus, the more feasible it's a chunk.

3) It's not determinate that how fine or coarse the chunk is. The most important thing is to avoid mistakes or generate the least wrong chunk alignments within aligned bilingual sentence pair.

4  Matching Algorithm

There are a variety of measures have been proposed for determining how closely a query is satisfied by a text in IR, such as Dice's coefficient, Cosine coefficient, etc. So it's possible to use them in a TM to find the similar source sentence of input sentence. However, it too coarse to find the most similar sentence for lacking the information of word orders. In the following section, the edit distance algorithm, in which the word order is included, is introduced and a user-guided matching algorithm is proposed.

4.1 Edit Distance

The edit distance algorithm is used to measure the string similarity by calculating the number of insertions, deletions, and substitutions to convert one string to another. The distance is obtained in \( m \times n \) operations (\( m \) and \( n \) is the character number of two strings being compared).

4.2 Multi-level Similar Segment Matching

Planas & Furuse[11] proposed a Multi-level Similar Segment Matching (MSSM) Algorithm based on Edit Distance Algorithm. The similarity is considered as a vector whose coordinates refer to the levels of analysis of the segments, such as
the surface word itself \((f=1)\), its lemma \((f=2)\), and its POS tag \((f=3)\). Here is an example taking from his paper [11]:

<table>
<thead>
<tr>
<th>C1</th>
<th>Sony</th>
<th>stay</th>
<th>ended</th>
<th>Monday</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>Sony</td>
<td>stay</td>
<td>end</td>
<td>Monday</td>
</tr>
<tr>
<td>C3</td>
<td>PN</td>
<td>noun</td>
<td>verb</td>
<td>Noun</td>
</tr>
</tbody>
</table>

Figure 2: An example of three-layer

By use of only the deletion and equality edit operations, the distance can be obtained in \(m^*\(n-m+1\) operations with the "lazy" match strategy, namely, if the first layer is same, the second and third layer are same too. In "exhaustive" match strategy, the distance can be obtained in \(m^*\(n-m+1\)^*f\) if there are \(f\) layers in consideration.

4.3 User-Guided Matching Algorithm

Different users need different kinds of help in translation. The reference sentence provided by system is a great help for some users, but it's a nuisance for others. Some users only need the system to find the exactly same sentence with the input sentence. Others may need the system to offer the sentences, which are similar to the input sentence in syntax structure, as a translation reference. If there are no considerations in matching algorithm of a system, it's difficult to offer the reference sentences according to user's need.

So in our system we proposed a user-guided three-layer matching algorithm to find what the user need. Here is an example:

IS: *He is a good student in our class.*
SS: *You are the best doctor in your hospital.*
TS: *你是最好的医生* (You are the best doctor in your hospital).

There is a TRP (Translation Relation Probability) table for the words in SS (Source Sentence) and the words in TS (Target Sentence) as show in table 2. The \(\Phi\) means there are no correspondence translation words found. In table 3a and 3b we show the edit distance algorithm used for system provided the user set the threshold value \(T_0\) is 0.7 and 0.3 respectively. The shade blocks of source sentence in table 3 mean the actually used layer for a word in similarity calculation. The similarity of two sentences changes from 62.5% to 100%. If only use the first-layer, namely the surface word, to match the similarity is only 11.1%.

Table 2: TRP table for the example

<table>
<thead>
<tr>
<th>SS</th>
<th>you</th>
<th>are</th>
<th>the</th>
<th>best</th>
<th>doctor</th>
<th>in</th>
<th>your</th>
<th>hospital</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td>你</td>
<td>是</td>
<td>(\Phi)</td>
<td>最好的</td>
<td>医生</td>
<td>(\Phi)</td>
<td>你们</td>
<td>医院</td>
</tr>
<tr>
<td>TRP</td>
<td>0.93</td>
<td>0.63</td>
<td>0.24</td>
<td>0.74</td>
<td>0.12</td>
<td>0.63</td>
<td>0.86</td>
<td></td>
</tr>
</tbody>
</table>

Table 3a: The similarity of two sentences (threshold \(T_0=0.7\))

<table>
<thead>
<tr>
<th>WORD</th>
<th>LEM</th>
<th>POS</th>
<th>be</th>
<th>he</th>
<th>is</th>
<th>a</th>
<th>a</th>
<th>good</th>
<th>student</th>
<th>in</th>
<th>our</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>You</td>
<td>you</td>
<td>pron</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>are</td>
<td>be</td>
<td>v</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>the</td>
<td>the</td>
<td>the</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>best</td>
<td>good</td>
<td>adj</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>doctor</td>
<td>doctor</td>
<td>N</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>in</td>
<td>prep</td>
<td></td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>your</td>
<td>your</td>
<td>pron</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>hospital</td>
<td>hospital</td>
<td>NOUN</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Table 3b: The similarity of two sentences (threshold \(T_0=0.3\))

<table>
<thead>
<tr>
<th>WORD</th>
<th>LEM</th>
<th>POS</th>
<th>be</th>
<th>he</th>
<th>is</th>
<th>a</th>
<th>a</th>
<th>good</th>
<th>student</th>
<th>in</th>
<th>our</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>you</td>
<td>you</td>
<td>pron</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>are</td>
<td>be</td>
<td>v</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>the</td>
<td>the</td>
<td>the</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>best</td>
<td>good</td>
<td>adj</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>doctor</td>
<td>doctor</td>
<td>N</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>in</td>
<td>prep</td>
<td></td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>your</td>
<td>your</td>
<td>pron</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>hospital</td>
<td>hospital</td>
<td>NOUN</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

930
The algorithm can be summarized as follows:

```plaintext
for j=1 to n  // n is the number of the source sentence
for i=1 to m  // m is the number of the input sentence
  if(TRP[j]i>Tj)f=3;  // f is layer level
  else if(TRP[j]i>Tj-Cj)f=2;
  else f=1;
  end if
  if(f[i]=Cj[i])  // equal in f layer
    d[i]=d[i-1, j-1]
  else d[i]=d[i-1, j-1]+1
    d[i, j]=min(d[i-1, j]+1, d[i, j-1]+1, d)
  end if
end for
end for
```

Figure 3 User-guided matching algorithm

As in complete lazy match in MESSM, the distance can be obtained in \( m \times (n-m+1) \) operations, if only the two operations, namely the deletions and equalities, is used.

5 Experimental Results & Discussion

We tested our system with an English-Chinese bilingual corpus, which is part of a computer handbook. There are about 1128 English sentences and 1063 Chinese sentences in this computer handbook after filter noisy figures and tables. Finally we extracted about 2729 different chunk pairs from the corpus. Here are three example sentences:

1E: [Once the file has been downloaded], [close your browser] [and all other] [background applications].
1C: [文件下载后], [关闭您的浏览器] [以及其他所有] [后台应用程序]。

2E: [Click the Finished button] [and then click Finished again] [to put your new Friends List into effect].
2C: [单击“完成”按钮], [然后再次单击“完成”按钮] [使新的“朋友列表”生效].

3E: [Sometimes errors occur during the installation procedure][that cause problems] [as you install the software] [or when you boot immediately after installation].
3C: [有时在安装过程发生的错误] [会使你在安装软件中] [或安装后立即引导时] [产生问题].

Figure 4: Three examples of Chunk alignment

The accuracy for automatic chunk alignment is 82.3%. The errors mainly due to the following reasons: Chinese segmentation error, stop words noise, POS tag error. With the help of system all the chunk alignments are corrected. The coverage to input sentence is estimated by following simple formula:

\[ \text{Cov.} = \frac{\text{Repeat No.}}{\text{Total No.}} \]

Where the numerator is the number of repeatedly appeared sentences or chunks in the experimental corpus, the denominator is the number of total sentences or chunks in the experimental corpus.

In Table 4 the Cov. of sentence and the Cov. of chunk are showed respectively. When sentence separated to chunk, the coverage is increased from 1.68% to 9.06%.

<table>
<thead>
<tr>
<th></th>
<th>Sentence</th>
<th>Chunk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeat Number</td>
<td>19</td>
<td>272</td>
</tr>
<tr>
<td>Total Number</td>
<td>1128</td>
<td>3001</td>
</tr>
<tr>
<td>Coverage</td>
<td>1.68%</td>
<td>9.06%</td>
</tr>
</tbody>
</table>

Table 4: The estimate of coverage

In figure 5 the complexities of the MESSM and the UGM (use-guided matching) are showed. The iterations of MESSM locate in the area between the curve 1 and 2 when lazy match during calculation. The iterations of MESSM is curve 2 when exhaustive match during calculation. The iterations of UGM are shown exactly in curve 1.

Figure 5: The iterations of MESSM & UGM

It can be see clearly from the curve that when \( m=(n+1)/2 \) the iteration number is maximum. So there are some ways to avoid this area and decrease the iteration time.
6 Conclusions

To use the bilingual corpora efficiently is a challenge task. With the development of hardwares of computer, it’s possible to propose complex algorithm to process huge corpora. At present, there are many research works about how to take advantage of the bilingual corpora and use them to machine translation or computer-aided as a knowledge base.

In this paper the pilot English Chinese Computer-Aided translation system based on bilingual corpora is described. The text alignment in corpus-processing module is based on chunk level in order to increase the coverage to input source sentence. The matching algorithm of input sentence and source sentence is a three-layer Edit Distance algorithm guided by user, which include the information of word morph and word POS. Preliminary experiments show promising results.

In future, some components of PECAT, such as the post-edit module need to be developed. The performance of all system should be evaluated from large-scale corpus.

Acknowledgement

Thanks to Emmanuel Planas for send his paper to me.

References