Portuguese-Chinese Machine Translation in Macao

Yiping Li
Faculty of Science and Technology
University of Macao
P.O.Box 3001, Macao

Chiman Pun
Faculty of Science and Technology
University of Macao
P.O.Box 3001, Macao

Fei Wu
Faculty of Science and Technology
University of Macao
P.O.Box 3001, Macao

Abstract

There have been substantial changes in computing practices in the cyberspace, mainly as a result of the proliferation of low priced under-utilized powerfully heterogeneous computers are connected by high-speed links. In this paper we reminisce the vicissitude of computing platform and introduce our Portuguese-Chinese corpus-based machine translation (CBMT) system which employs a statistical approach with automatic bilingual alignment support. Our improved algorithm for aligning bilingual parallel texts can achieve 97% of accuracy. At the same time, we broach the "distributed translation computing" concept to construct a uniform distributed shared-object technical term retrieving workstation and achieve high computing performance balance of network where heterogeneous computers inherently root and are intermittently under-utilized. Whereby it, we can expedite to retrieve technical terms from noisy bilingual web text and build up the Portuguese-Chinese corpus-base.

1 The vicissitude of computing platforms

Automatic translation between human languages ('Machine Translation') was one of the earliest tantalizing applications suggested for digital computers, but turning this dream into reality has turned out to be a much harder, and in many ways a much more interesting task than at first appeared.

If machine translation could be regarded as the one of earliest complicated synthetic computation in cyberspace, then where the computing concept stems? Two thousand years ago, Euclid set a famous standard for rigor in geometrical proofs, which is called Completeness and Consistency now. In order to settle the standard initiated by Euclid, Logicians and mathematicians have to first conceive an effective computing platform to determine whether a given proposed inference is valid from the premises. There have ever been many examples of innovative concepts and computing platforms introduced by logicians and mathematicians which later proved to be important in current computer science, but all of those computing platforms are immanently confined to certain cases.

In 1936 papers by Church, Kleene, Turing and Post were almost simultaneously published. Each of the authors proposed a precise definition for the seemingly vague notion of effective computing platform, namely recursive function (Godel- Herbrand, 1934), Lemada-Calculus (Church-Kleene, 1932-1934), Turing machine (Alan Turing 1936). According to Church thesis, all of computing platforms above can be shown to be equivalent. It is not uncommon in mathematics and in the other sciences for concepts, methods, and theorems to be discovered independently and almost at the same time. Why eventually Turing's work has proved to be of greatest mathematical and philosophical significance than that of the other authors and adumbrates the first powerful all-purpose digital computer? What Turing did was to show that calculation can be broken down into the iteration (controlled by an internal state program) of extremely simple concrete operations; the processes of calculation is so concrete that they can be easily described in terms of physical mechanisms. However, the computing platforms proposed by other logicians and mathematicians were based on mathematical and logical experience of that time. Even though incredible, the first effective computing platform debuts: the computing platform can be viewed as a sophisticated tape player, with an arbitrarily extendable tape. The tape is marked off a series of squares, each of which can hold a single symbol. The tape head, or read-write head, can read a symbol from the tape, write a symbol to the tape, and move one square in either direction at a time. The model of Turing computing platform directly leads to the germination of world first digital computer ENIAC.

The memory limitation of machines and inferior compilers are inevitable bottleneck for early computers to achieve higher computing performance. As a result, earlier computing platforms including machine translation are mainframe-centric because mainframes have much more memory and higher executing speed than PCs do.

In the late 1960s and early 1970s, people realized that software costs were booming faster than hardware cost. What makes it unbearably is programmers at that time could never guarantee the correctness of their

software. Many large software projects are over-budget and behind the schedule. In order to handle "software crisis", highly microcoded computer platforms were invented. The aim of microcoded architecture is to provide much more high hardware-level instructions to programmers and reduce the semantic gap between programmers and naked machines. However, The exertion did not make the mainframe-centric computing platform faded at all and resolve "software crisis" indeed. From 1980s, the performance of personal computers approximates to that of mainframe gradually. As a result, the pendulum swung from mainframe host-based to desktop computing.

When networks became ubiquitous. Many companies in the middle 1990s were completing their moves to client/server platform comprising a very fat client in which most of the end-user interaction occurs on PCs and a server dedicated to either file or database access. Many had found, however, that the supposedly cheaper and more powerful client/server platforms were actually more expensive to build, more difficult to maintain. IT professionals felt that they had to replace computers and networks, and install new versions of software on an intermittent basis. IT professionals were beginning to feel that they were on IT treadmill, moving faster and faster with no "off" switch.

The problems had been brewing for a couple of years. However, when web browser is becoming the preferred data delivery mechanism across global community in the late of 1990s, a new computing platform is sprouting to cope with this impasse: the thin client/ network computing platform, which incarnates the Sun's moving slogan "network is the computer".

Advances in network are making networked computing the most common form of high performance computing (HPC) now. Networked computing perhaps makes us to move away from a windowing environment, which mimics a physical desktop. The most important pending development in HPC, in our opinion, is not the creation of dedicated and fast new supercomputers clustered together. Instead, it is the individually heterogeneous computers co-operate to cope with a problem in a co-operative manner over certain geographical distance.

As a result, making machine translation to cater for the novel changes in HPC is one of indisputable frontier in MT realm. In part 3, we show how our technical retrieving term workstation satisfies these novel changes in HPC, that is to say, connecting heterogeneous computers to coordinate them for a cooperative work.

2 Portuguese-Chinese corpus-based machine translation system

The 1990s have witnessed a resurgence of interest in empirical and statistical method in machine translation. Perhaps the immediate reason for the empirical renaissance is the availability of massive quantities of data from web.

Our PCs standalone Portuguese-Chinese corpus based machine translation (CBMT) is a statistical-based approach to perform automatic machine translation by means of retrieving translation units, such as words, phrases, or sentences, from a large bilingual corpus-base with possibly regular and irregular transformations.

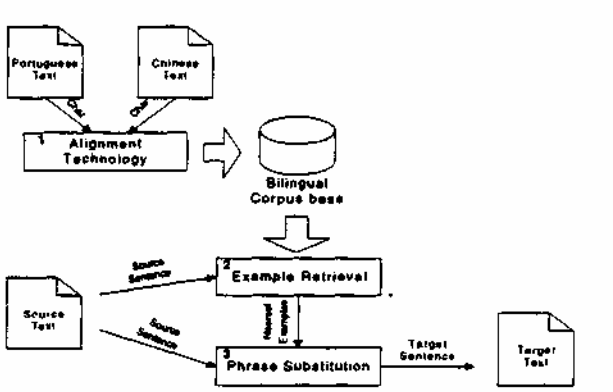

Figure 1 Process of CBMT

The idea of CBMT is actually deduced from example-based machine translation (EBMT). The core part of CBMT is, of course, the bilingual corpus-base, like the bilingual examples in EBMT. In fact, both CBMT and EBMT share almost the same theory, except that their principal translation units are distinct. In CBMT, basic translation units for the bilingual corpus-base can be sentences, phrases or words.

The conventional rule-based machine translation (RBMT) fails due to the lack of machine-readable, formal and complete descriptions of the human languages. The availability of a large amount of bilingual Portuguese-Chinese text in Macao nowadays, solves the one of major problems in CBMT, namely, finding the intensive data source of bilingual texts for the Portuguese-Chinese corpus-base.

There are mainly three key issues in CBMT (Fig. 1):

1. Building the bilingual corpus-base at sentence, phrases and word level from Portuguese-Chinese texts.
2. A mechanism for retrieving the example that best matches the input sentence from the bilingual corpus-base.
3. Exploiting the retrieved translation units to produce the actual translation of the input sentence.

The Alignment Technology in Figure 1 is a segmentation of the bilingual texts, typically into small logical units such as sentences and words, such that the n^{th} segment of the first text and the n^{th} segment of the second are mutual translations.

Approaches to sentence alignment basically fall into two diverse classes: lexical and statistical. The lexically-based method uses a lot of online bilingual lexicons and heuristic linguistic knowledge to match sentences. The statistical-based methods, on the other hand, usually require no or very little linguistic knowledge and are solely based on the lengths of parallel sentences. Because the exhaustive-searching heuristic method is intrinsically prone to be a NP-hard problem, The NP-irrelevant advantage of later pragmatic approach makes it prevailing.

Given a pair of parallel Portuguese-Chinese texts (passages), Alignment chooses the corresponding alignment that maximizes the probability over all possible alignments. Formally,

$$\arg\max_{A} \text{Prob}(A \mid T_1, T_2) \qquad (1)$$

where A is an alignment, and T_1 and T_2 are the Portuguese and Chinese texts, respectively. An alignment A is a set consisting of $L_1 \leftrightarrow L_2$ pairs where each L_1 or L_2 is a Portuguese or Chinese passage.

We can now do some approximations so that the formulation is not too general. Assume that the probabilities of the individually aligned pairs within an alignment are independent, that is, A_i does not depend on A_j for any $i \neq j$, then the first approximation is as follow:

$$\text{Prob}(A \mid T_1, T_2) \approx \prod_{(L_1 \leftrightarrow L_2) \in A} \text{Prob}(L_1 \leftrightarrow L_2 \mid T_1, T_2) \qquad (2)$$

Usually each Prob ($L \leftrightarrow L_2 \mid T_1, T_2$) does not depend on the entire texts, but only on the contents of the specific passages within the alignment, then we have the second approximation as follow:

$$\text{Prob}(A \mid T_1, T_2) \approx \prod_{(L_1 \leftrightarrow L_2) \in A} \text{Prob}(L_1 \leftrightarrow L_2 \mid L_1, L_2) \qquad (3)$$

From (1), the maximization of the approximation (3) to the alignment probabilities is easily converted into a minimum-sum problem:

$$\begin{aligned} &\arg\max_{A} \text{Prob}(A \mid T_1, T_2) \\ &= \arg\max_{A} \prod_{(L_1 \leftrightarrow L_2) \in A} \text{Prob}(L_1 \leftrightarrow L_2 \mid L_1, L_2) \\ &= \arg\min_{A} \sum_{(L_1 \leftrightarrow L_2) \in A} -\log \text{Prob}(L_1 \leftrightarrow L_2 \mid L_1, L_2) \end{aligned} \qquad (4)$$

The *log* is introduced here so that adding the probabilities will produce desirable results.

Many alignment methods exclusively employed statistical criteria, like Gale and Church's pure length-based alignment method [Gale91]. Some other alignment methods solely employ lexical criteria. Only few, like Wu's, combine both statistical and lexical criteria in sentence alignment [Wu95]. However, the performance would be egregiously degraded when the lexical database becomes trivially larger. Our initiatives are to incorporate lexical criteria without compromising the high performance obtained by statistical approach.

Pure length-based method is based on the fact that longer sentences in source language tend to be translated into longer sentences in the target language, and that shorter sentences tend to be translated into shorter sentences. From (3), we can rewrite the probability function as follow:

$$\text{Prob}(L_1 \leftrightarrow L_2 \mid L_1, L_2) \approx \text{Prob}(L_1 \leftrightarrow L_2 \mid l_1, l_2) \qquad (5$$

In order to incorporate lexical criteria into the pure length-based method, we include some lexical parameters in (5).

$$\begin{aligned} &\text{Prob}(L_1 \leftrightarrow L_2 \mid L_1, L_2) \\ &\approx \text{Prob}(L_1 \leftrightarrow L_2 \mid l_1, l_2, v_1, w_2, \ldots, v_n, w_n) \end{aligned} \qquad (6$$

Where l_1 = the length of L_1 and l_2 = the length of L_2, measured in number of characters, v_i = the number of Portuguese cue of $type_i$ in L_1 and w_i = the number of Chinese cue of $type_i$ in L_2 (see Table 1).

The following is the criteria for choosing the lexical cues:

1. should be highly reliable and occur frequently so that violations, which wastes the additional computation, happen only rarely;
2. domain specific lexical cues are useful for building domain specific corpus-base;
3. both Portuguese and Chinese fields of a lexical cue should be unique.

In similar way of [Gale91], the dependence can be encapsulated by different parameter δ_i, and the $L_1 \leftrightarrow L_2$ pairs in alignment set can also be restricted into six matches: 1-1, 0-1 or 1-0, 2-1 or 1-2, 2-2 mappings. The approximation is estimated as follow:

$$\begin{aligned} &\text{Prob}(L_1 \leftrightarrow L_2 \mid l_1, l_2, v_1, w_2, \ldots, v_n, w_n) \\ &\approx \text{Prob}(\textit{match} \mid \delta_0(l_1, l_2), \delta_1(v_1, w_1), \ldots, \delta_n(v_n, w_n)) \end{aligned} \qquad (7$$

Table 1 Some examples of Portuguese-Chinese lexical cues

Cue No.	Type	Portuguese	Chinese
1	adj.	1	一
2	adj.	2	二
3	adj.	3	三
¡ K		¡ K	¡ K
N	noun	Agosto	八月

By Bayes' Theorem, we have

$$\text{Prob}(match \mid \delta_0,\delta_1,\ldots,\delta_n) = \frac{\text{Prob}(\delta_0,\delta_1,\ldots,\delta_n \mid match)\,\text{Prob}(match)}{\text{Prob}(\delta_0,\delta_1,\ldots,\delta_n)} \propto \text{Prob}(\delta_0,\delta_1,\ldots,\delta_n \mid match)\,\text{Prob}(match) \quad (8)$$

Where $Prob(\delta_0, \delta_1, {}_j K\delta_n)$ is a normalizing constant and can be ignored. It would not affect the result of the minimization because $\delta_0, \delta_1, {}_j K\delta_n$ do not depend on *match.*

Assuming all δ_i values are approximately independent, we have,

$$\text{Prob}(\delta_0,\delta_1,\ldots,\delta_n \mid match) \approx \prod_{i=0}^{n} \text{Prob}(\delta_i \mid match) \quad (9)$$

Similarly, we follow the definition of the function δ in [Gale91], since it has a approximately normal distribution with mean zero and variance one:

$$\delta_0 = \frac{l_2 - l_1 c}{\sqrt{l_1 s^2}} \quad (10)$$

Where l_1 = the length of L_1 (Portuguese Sentence) and l_2 = the length of L_2 (Chinese Sentence), c is the expected number of characters in L_2 per character in L_1, and s^2 is the variance of the number of characters in L_2 per character in L_1. Here, the mean, c, and variance, s^2, have exactly the same values for paragraph and sentence alignment.

And,

$$\delta_i = \frac{w_i - v_i m}{\sqrt{v_i v^2}}, \quad i = 1..n \quad (11)$$

Where v_i = the number of Portuguese cue of $type_i$ in L_1 (Portuguese Sentence) and w_i = the number of Chinese cue of $type_i$ in L_2 (Chinese Sentence), m is the mean of the number of Chinese cue of $type_i$ in L_2 per number of Portuguese cue of $type_i$ in L_1, and v^2 is the variance of the number of Chinese cue of $type_i$ in L_2 per number of Portuguese cue of $type_i$ in L_1.

The mean and variance calculated in [Gale91] are primarily for English-French and English-German alignment. We have to work out our own mean and variance for Portuguese-Chinese alignment. Based on the data of our own Portuguese-Chinese parallel texts in [6], We define our own mean and variance as follow:

c (mean) = 0.694
s^2 (variance) = 0.204

In addition, the number of Portuguese cue of $type_i$ and the number of Chinese cue of $type_i$ are most probably one-to-one mapping, that is, if there is only one occurrence of Portuguese cue in a particular Portuguese sentence, then most probably there should have only one occurrence of Chinese cue in the corresponding Chinese sentence. Hence, we define the nearly standard values for the mean and variance as:

m (mean) = 0.9
v^2 (variance) = 0.01

Table2 The probability of the six matches in sentences

Category	Prob(match)
1-1	0.899
1-0 or 0-1	0.0099
2-1 or 1-2	0.089
2-2	0.011

Similarly to[Gale91], we use the same probability table for *Prob(match)* as described in previous section. Hence, the conditional probability $Prob(\delta_i \mid match)$ can also be estimated in the same way by:

$$Prob(\delta_i \mid match) = 2(1 - Prob(|\delta_i|)) \quad (12)$$

Where $Prob(|\delta_i|)$ is the probability that a random variable, z, with a standardized normal distribution, has magnitude at least as large as $|\delta_i|$. That is,

$$\text{Prob}(\delta_i) = \frac{1}{\sqrt{2\pi}} \int_{-\delta_i}^{\delta_i} e^{-z^2/2}\,dz, \quad i = 0..n \quad (13)$$

Finally, the completed formula to select suitable alignment pairs for length-base sentence alignment with the incorporation of lexical cues can be obtained by (4), (9), (12), (13) as follow:

$$\begin{aligned}
&\arg\min_A \sum_{(L_1 \leftrightarrow L_2) \in A} -\log \text{Prob}(L_1 \leftrightarrow L_2 \mid L_1, L_2) \\
&= \arg\min_A \sum_{match \in A} -\log \text{Prob}(\delta_0,\delta_1,\ldots,\delta_n \mid \text{match})\,\text{Prob}(\text{match}) \\
&\approx \arg\min_A \sum_{match \in A} -\log \prod_{i=0}^{n} (2(1 - \text{Prob}(|\delta_i|))\,\text{Prob}(\text{match}) \\
&\approx \arg\min_A \sum_{match \in A} -\log \prod_{i=0}^{n} \left(2\left(1 - \frac{1}{\sqrt{2\pi}} \int_{-\delta_i}^{\delta_i} e^{-z^2/2}\,dz\right)\right) \text{Prob}(\text{match})
\end{aligned} \quad (14)$$

Where δ_i and *Prob(match)* are defined in (10, 11) and Table2, respectively.

In corpus-based machine translation, the underlying translation mechanism is to find out the nearest example matching the being translated input sentence from bilingual corpus-base. A nearest example of an input

sentence is the one which has the highest score on the metric of similarity (as defined below) after the search in the bilingual corpus-base with regular and irregular transformations.

Definition of Similarity Metric is given below:

- the comparison units can be sentences, phrases or words, giving higher score to longer unit;
- if the functional words in both sentences fall into the same classes, that is, they have same pure form, after the regular and irregular transformation, then this pair has higher score than different classes.
- the minimum difference in the length of two sentences will have higher score.

In order to improve the quality of raw target translation text, in CBMT, we provide a user-friendly corpus-base building environment for human translator's post-editing work as the strong system back-end. Translators implement the phrase substitution function to replace phrase with its most potential substitution.

We evaluate the alignment algorithm in CBMT. The alignment algorithm is a length-based method with incorporation of lexical cues. It has two steps: aligning the paragraphs first and then sentences. Although these two steps employ the same algorithm except the parameters: the variance and mean are different, they can produce very different results because of the differences in distributions between paragraphs and sentences (as shown Table3 & Table4). The aligned results are compared with a human alignment to decide whether a particular pair of alignment is correct or not. The task of human alignment is very simple, which is mostly to identify the wrong sentence segmentation and minor linguistic ambiguity.

Table 3 Results of sentence alignment

	1-1	1-0	0-1	2-1	1-2	2-2
Total	384	0	0	35	3	2
Correct	380	0	0	33	1	2
Incorrect	4	0	0	2	2	0
% Correct	0.99			0.94	0.33	

Table 4 Results of paragraph alignment

	1-1	1-0	0-1	2-1	1-2	2-2
Total	411	0	0	3	3	0
Correct	411	0	0	2	2	0
Incorrect	0	0	0	1	1	0
% Correct	100			66.7	66.7	

Our experiments of CBMT have produced encouraging results in alignment algorithm, which can have 97% of accuracy in sentence alignment and 99% of accuracy in paragraph alignment. From the result of sentence alignment, most of aligned pairs are 1-1 mapping and its error rate is only about 1%. The main reason for these results is the high frequency of 1-1 alignments in Portuguese-Chinese translations. There are no examples in 1-0 and 0-1 mappings because of the extremely low probability in these two cases and in fact there are no such sentences in the original texts. The result of high error rate in the 2-1 mappings is due to the specially structure in original texts and different punctuation between them (sometimes the Portuguese text uses full stops to separate numbering header, sometimes uses parentheses). Generally, the error rate depends mostly on following four factors:

1. Sentence Length
2. Paragraph Length
3. Category Type
4. Probability Measure

Although the alignment algorithm is very accurate and fast, the aligned results are not very useful in automatic translation, because most of sentences are not concise enough for phrase substitution algorithm. One obvious improvement is based on the results of sentence alignment to do word alignment. Because normal pairs of Portuguese-Chinese words have been comprised by dictionary, aligning these words are insignificant. However, retrieving technical terms to build up bilingual Portuguese-Chinese corpus-base is solicitous. This topic is discussed in part 3.

3 A distributed shared-object technical term retrieving workstation

Technical-term translation represents one of the most difficult tasks for machine translation. Since such domain-specific terminology is not adequately covered by printed dictionaries. Retrieving pairs of Portuguese-Chinese technical terms from bilingual texts to build up the bilingual corpus-base is an enormously cumbersome work because HanZi (Chinese Character set) is not delimited by space like English. However, many nomenclature terms can not be found in pocket dictionaries but referenced with high frequency in specific domain. Encompassing much more those technical terms in corpus-base would expedite translation process. After web becomes a prime vehicle for disseminating information in cyberspace, many bilingual hypertext which are translations of each other but are not translated sentence by sentence are put into web. Unfortunately, these online hypertext have many mismatched sentences and paragraphs, suggesting a discontinuous mapping between bilingual hypertexts. Most of segmentation algorithm including the alignment algorithm in above-mentioned CBMT can not deal with deletions and insertions in bilingual hypertexts efficiently and therefore can not be applied to such noisy bilingual web hypertexts.

Furthermore, As said before, the blossoming of networked computing galvanizes us to construct a platform to both cater the pending platform revolution and achieve high performance computing in local network when retrieving domain-specific jargon from web.

Basically speaking, there are two platforms to make high performance computing (HPC) viable, namely multi-computer and multi-processor. A multi-processor is a machine with multiple CPUs sharing a single common virtual address space. All CPUs can read and write every location in this address space. Multi-processor can be programmed using well-established techniques, but they are difficult and prohibitive to build. For this reason, many multi-processor systems are simply a collection of independence CPU-memory pairs, connected by a communication network. Machines of this type that do not share primary memory are called multi-computer. Because multi-computer is easier to build and is likely to monopolize the HPC market in the future, multi-computer is heatedly researched.

The usual approach to programming multi-computer is message passing. The operating system provides primitives SEND and RECEIVE in one form or another, and programmers can use these for interprocess communication. This makes I/O the central paradigm for multi-computer software, something that is unfamiliar and unnatural for many programmers.

An alternative approach is to simulate shared memory on multi-computer. In this system, a collection of workstations on a local area network share a single, paged, virtual address space. The pages are distributed among the workstations. When a CPU references a page that is not present locally, it gets a page fault. The page fault handler then determines which CPU replies by sending the page. Although various optimizations are possible, the performance of this system is often inadequate.

Our endeavor of obtaining HPC is based on the concept of parallel retrieving technical terms using shared-object from noisy web text through CORBA in heterogeneous environments when any under-utilized computer is available, that is, constructing a distributed shared-object technical term retrieving workstation.

Executing shared objects has three advantages over preceding two HPC platforms. First, the data is encapsulated inside object; second, the shared object can be done locally or globally on any machine which is interconnected no matter where the shared object is stored and no matter which operating system the machine runs; finally, more parallelism is possible since multiple machines can be calling and running the same shared object at the same time.

In the distributed shared-object technical term retrieving workstation, we store the shared object in an object-server. It would be called through CORBA by other applications running at clients, even through the applications of callee and caller are written by different system language, and clients and object-server are based on different operating systems, character coding schema and network protocols etc! This is a remarkably hectic advantage of CORBA —"Cross-Language"!

CORBA is defined in late 1990. Instead of attempting to enforce universality along any of the dimensions, CORBA presents three critical components for application integration in distributed, heterogeneous systems:

1.an object-oriented application programming interface specification languages, OMG Interface Definition Language (OMG IDL), separate from any programming language;

2.a set of application programming interfaces for discovering and accessing application services on any reachable heterogeneous platform; and

3.a platform-independent interpretability framework including TCP/IP-based protocol (called Internet Inter-ORB protocol Or CORBA HOP)

Through the three components supplied by CORBA, Objects can be performed on different machines as through they were in physical shared memory.

In the distributed shared-object technical term retrieving workstation, If there is any under-utilized computer, the under-utilized computer remotely or locally calls technical term retrieving object stored in object-server and engages in retrieving technical terms. After getting the possible pairs of Portuguese-Chinese technical terms, the under-utilized computer sends retrieved results to object-server, object-server inserts the results into corpus-base at last. Whereby it, the computing utilization in local network can achieve full-utilization and balance. This is the reason why we call "Distributed Computing Translation".

After getting the technical term retrieving object from object-server, the client uses the dynamic time-warping (DTW) algorithm to retrieve the pairs of technical terms from web. According to Pascale Fung and Kathleen Mckeown [1997], when similar words in noisy web bilingual hypertext do not occur at the exact same position in each half of the bilingual hypertexts, distances between instances of the same word will be similar across languages. Hence, retrieving technical terms is converted into a pattern-matching problem—each word shares some common features with its counterpart in the bilingual hypertexts.

Because the similarity between patterns can be measured in a mathematically rigorous manner if the patterns are represented in a vector space, we represent the occurrence information of the words in vector form. The matched bilingual technical words may not necessarily occur within the same segments, nor do they occur at the same byte positions in the corresponding hypertexts.

If we define "arrival interval" to be the difference

between successive byte positions of a word in the hypertext, the arrival intervals are more similar between word pairs than their absolute positions. This arrival interval can be regarded as the recency information of words and can be used as a feature in the pattern matching of word pairs.

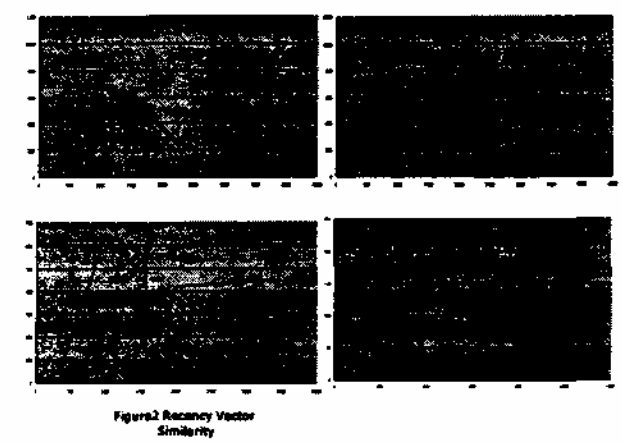

Figure2 Recency Vector Similarity

Figuratively, for each words, we have a signal such as is shown in Figure with the horizontal axis representing the word position in the hypertext and the vertical axis the values of the recency vectors. The striking similarity of the two signals which represent Governor of Special Administration in Portuguese (Upper left in Figure 2) and Chinese (Upper right in Figure 2) corroborates our assumption. The signal for the word Casino (Bottom right in Figure 2) is clearly different from that of Governor of Special Administration in Portuguese and Chinese. Since Macao is a famous casino showplace and Hotel Lisbon is a notable lieu providing casino service, Casino and Hotel Lisbon are nearly endemic interchange in Macao. However, the signal for Hotel Lisbon (Bottom right in Figure 2) is also significantly different from that of Hotel Lisbon.

Looking at the values of individual recency vectors, it is hard for us to discern which pairs are indeed similar to each other. However, the signals show that each vector has a characteristic shape which can be used for pattern matching and distinguishing translation of technical terms from noisy web bilingual hypertexts.

The recency vector representation assumes that if two technical terms are translations of each other, they are more likely to occur a similar number of times at similar arrival intervals.

Since we do not apply any word demarcation algorithm to web hypertext before retrieving technical terms, we use speculation mechanism here. First, We randomly choose any length of one term in source web hypertext as one possible technical term; Second, we try to find which length of one term in corresponding target hypertext matches the source term in recency vector representation; third, if such matching metric approaches to the criterion set up formerly then algorithm terminates, otherwise we decrease or increase the length of source term one by one to iterate the matching algorithm until the matching pair is found.

Our one-step DTW algorithm calculation is as follows:

Initialization

$£(1,1) = ¢(1, 1)$
$£(i,1) = ¢(i, 1) + ¢(i-1, 1)$
$£(i,j) = ¢(1, j) + ¢(1, j-1)$
where $¢(a, b) = |V1[a]-V2[b]|$
for i=2,3,..., N
j= 2,3...,M
N=dim(V1)
M=dim(V2)

V1 and V2 are recency vectors of terms being matched

Recursion

The accumulated cost £ of the DTM path is defined recursively:

$£(i,j)=\min\{£(i-1,j)+¢(i,j), £(i-1,j-1)+¢(i,j), £(i,j-1)+¢(i,j)\}$
$¢(i,j)=1$ if $£(i,j)=£(i-1,j)+¢(i,j)$
$¢(i,j)=2$ if $£(i,j)=£(i-1,j-1)+¢(i,j)$
$¢(i,j)=3$ if $£(i,j)=£(i,j-1)+¢(i,j)$

Termination

$£(N,M)=\min\{£(N-1,M)+¢(N,M), £(N-1,M-1)+¢(N,M), £(N,M-1)+¢(N,M)\}$
$¢(N,M)=1$ if $£(N,M)=£(N-1,M)+¢(N,M)$
$¢(N,M)=2$ if $£(N,M)=£(N-1,M-1)+¢(N,M)$
$¢(N,M)=3$ if $£(N,M)=£(N,M-1)+¢(N,M)$

Final cost of the DTM path is normalized by the length of the path:DTM (N,M)= £(N, M)/(N+M)

If the final cost reaches the criterion set up previously, one pair of technical terms is retrieved. Otherwise we change the length of source term and repeat the DTM algorithm until a pair of nearly same distribution of technical term is found. Although speculation enormously increases complexity, however the DTM algorithm is **Turing tractable** rather than **NP-Complete** since the complexity is O(length(Source hypertext) * Length (Target hypertext) in its worst case.

The reason why we use distributed computing translation platform is clear now: building up our corpus-base by fully using under-utilized computers is one point but not all; constructing a uniform platform in heterogeneous cyberspace to glue their inherently incompatible features and achieve the fully computing

balance in local network are the key point. Because the shared object realizes certain parallel applications on systems lacking physical or logical conformity, it is a future paradigm in heterogeneous cyberspace such as Macao.

4 Future Remarks

From the theoretical Turing computing platform to the first embryonic digital computer ENIAC platform, from Mainframe-centric platform to PC-centric platform and current Network computing platform, even though the performance of computers is prodigiously enhanced, achieving high performance computing power is our human being uninterruptedly gangling endeavor including machine translation field.

For over a decade prophets have voiced the contention that the organization of a single computer has reached its limits and that truly significant advances can be made only by interconnection of a multiplicity of computers in such a manners as to permit cooperative solution. However, the dream of building computers by simply aggregating processors has been checkered because rarity of parallel software environments especially in heterogeneous cyberspace where the diverse computers root. In this paper, we introduce our distributed shared-object technical term retrieving workstation and CBMT. The former can obtain high computing performance and optimize the computing performance balance in local network where heterogeneous under-utilized computers aggregate. The later uses the static and rule based method to obtain a sound alignment result.

References

Pascale Fung &Kathleen Mckeown (1997), A Technical Word- and Term-Translation Aid Using Noisy Parallel Corpora across Language Groups, Machine Translation 12

Gale, William. A. & Kenneth W. Church (1993), A Program for Aligning Sentences in Bilingual Corpora, *in Using Large Corpora,* The MIT Press

KWONG-SAK LEUNG, KIN-HONG LEE AND YUK-YIN WONG (1998), DJM:A Global Distributed Virtual Machine on the Internet, Software-Practice and Experience Vol.28(12)

Pun, Chi Man (1997), *A Portuguese-Chinese Corpus-Based Machine Translation,* Thesis Report, University of Macao

Wu Fei (1999), A domain-based Portuguese-Chinese Online machine translation system, Thesis Report, University of Macao

Rolf Herken (Ed) (1995), The Universal Turing Machine A Half-Century Survey (Second Edition), New York, Springer-Verlag Wien

Li Yi-Ping & Wu Fei (1998), Advances of On-line Machine Translation, Chinese Translators Journal, Vol. 127 No.l

Li, Yiping, Pun, Chiman & Wu, Fei (1999), Online Multilingual Computing and Portuguese-Chinese MT, Proceedings of International Symposium on Machine Translation & Computer Language Information Processing (ISMT&CLIP), Beijing, P.R.China

Li Yiping, Wu Fei (1999), Distributed Extracting Technical Terms from Multilingual Cyberspace, Journal of LanZhou University (Natural Science), Vol.35

Church, Kenneth W. and Robert L. Mercer (1993), Introduction to the Special Issue on Computational Linguistics Using Large Corpora, *in Using Large Corpora,* The MIT Press