自然言語処理 101－7
（1994. 5. 27）

漢字対応の利用による日中対訳テキストの文対応付け

陳 樹霖[*]　長尾 眞[†]
[*] 国立シンガポール大学 情報理学系
[†] 京都大学工学部 電気工学第二教室

要旨

日本語と中国語の対訳テキスト中には、共通の漢字が現れるのという特徴がある。そのため、一方の言語しか理解できない読者でも、日中対訳テキストの文対応付けを行なうことがある程度可能である。本論文では、この特徴を語彙的制限として利用することによって、計算機による日中対訳テキストの文対応付けの精度を改良する方法について述べる。

Kanji as Lexical Constraints in Alignment of Japanese and Chinese Texts

Chew Lim Tan[*]　Makoto Nagao[†]
[*] Department of Information Systems & Computer Science, National University of Singapore
[†] Department of Electrical Engineering, Kyoto University

Abstract

An interesting and unique feature in the Japanese and Chinese language pair is the occurrence of common kanji or Chinese characters in parallel texts which in fact allows even a monolingual reader to do a partial alignment of corresponding sentences from the parallel texts. This paper describes how this feature may be used as lexical constraints to improve the accuracy of automatic alignment of Japanese and Chinese texts.

1 Introduction

Japanese and Chinese are linguistically far apart from each other. Yet there is a distinct commonality between the language pair, that is the use of kanji or Chinese characters which allows even a monolingual Chinese reader to have a partial understanding of Japanese texts. Given a parallel Japanese-Chinese translation texts, the monolingual reader can also manage a rough alignment of corresponding sentences from the bilingual texts. The degree of accuracy of alignment of course depends on the extent of common Chinese characters used. The reader will therefore have no way of aligning texts that contain only hiragana and katakana such as those in young children literature. However, general documents and especially technical articles have enough kanji to permit reasonable alignment by hand. Alignment by hand, however, is necessarily a laborious task. It will be interesting to see how Chinese characters in the bilingual texts could be of use to automatic alignment in the same way as in manual alignment.

The study of automatic alignment has in recent years received attention from several researchers [1][2][3][4][5]. It is particularly useful to example-based machine translation proposed by Nagao[6] which requires a huge volume of translation example pairs. Work on automatic alignment reported in the literature may be classified into two main categories, one being lexically based and the other statistically based. The lexically based approach as in [1][2][5] relies on the lexical contents of prospective sentence pairs. For instance, in the work of Hwang and Nagao[5] for aligning Japanese-Korean bilingual texts, words in each Japanese sentence are first converted to Korean words by consulting some lexicons. An attempt is then made to find a Korean sentence in the text that has the best match with the converted sentence. Another lexically based approach by Kay and Roscheisen[2] attempts to identify association of words between the two texts based on the similarity of word distributions in the texts. The statistically based approach, however, performs alignment merely by comparing sentence lengths of potentially matching pairs with no regards to their lexical contents at all. It is based on an intuitive premise that longer sentences tend to be translated into longer sentences in the target language and that shorter sentences tend to be translated into shorter sentences. The methods by Brown *et al.*[3] and by Gale and Church[4] fall into this category but Brown *et al.* measure sentence lengths in words while Gale and Church use characters as the basis of measure.

Our work here represents a combination of the above two approaches by allowing occurrence of kanji to assist a statistical alignment based on sentence lengths. Because of the lack of word delimiters in both Chinese and Japanese texts, we will use characters (rather than words) as the measure of sentence lengths as in the work of Gale and Church[4]. In fact, Gale and Church observed that their statistically based method could achieve a high accuracy of 96% and suggested that the remaining errors could be remedied by introducing lexical constraints in their future work. The use of kanji occurrence in our work offers a unique yet simple means of lexical constraints not possible in other language pairs. Our approach will now be described. Section 2 will first demonstrate the need for kanji as lexical constraints. Section 3 will then discuss the basic strategy of matching. Section 4 will then present the alignment algorithm in detail using mathematical notations. Section 5 will give the results of our experiments. Finally, section 6 will conclude this paper with a brief discussion of our future work.

2 The Issues of Kanji

An issue that we need to address is that although common kanji are found in parallel Chinese and Japanese texts, different coding systems are actually used for handling both texts which render them totally different to the computer. A public domain software *sinocode* helps to solve the problem by allowing conversion among different coding systems, such as JIS, GB, and Big5. As we are working in the kterm (Kanji terminal emulator) environment, all our Chinese texts are converted to JIS code. Thus in this paper, all Chinese characters in the Chinese texts appear in their corresponding kanji. Chinese characters that have no corresponding JIS code are converted to a special symbol by sinocode.

Another issue that we want to examine was how much useful the reliance on kanji would be, as the pure statistical approach by Gale and Church[4] has already achieved a high accuracy of 96%. Gale and Church also believe that their method should work equally well on other language pairs. While in theory the claim appears true as the method has no reliance on lexical contents at all, a closer examination of parallel texts of Chinese and Japanese reveals two problems not found in the earlier work. The first arises from an observation that Japanese words may be written in kanji or hiragana. Thus “皆さん、今日は” and “みなさん、こんにちは” are just the same and may be translated to Chinese as “各位好”. This gives rise to a wider variability in the ratio for the number of Japanese characters to that of Chinese characters. A rough sampling over some 60 bilingual texts from such bilingual materials as general essays[7][8][9], news reports[10], scientific articles[11][12][13], and language lessons[14], shows that while most parallel sentences give an average ratio of 1.4 for the number of Japanese characters (kanji/hiragana/katakana) over each Chinese character translated, the ratio can range from 0.5 to 4. The wider variance will thus give rise to a greater error in doing matching based on statistical measure.

The other problem comes from differences in sentential structures between the two languages. It is observed that the punctuation mark period (“。”) may be used at different locations in the bilingual texts thus preventing a simple matching at the sentence level. The earlier works on alignment [1][2][3][4][5] have all required that sentences be matched as a whole. Gale and Church[4], for instance, allow matching between different combination of sentences from zero up to two sentences but in the case of Japanese and Chinese bilingual texts, matching between a greater multiple number of sentences must be catered for. We have found in one example that one Japanese sentence has to be matched with 5 Chinese sentences. Allowing a wider range of matching between sentences, however, will greatly reduce the efficiency of the alignment algorithm. Furthermore, the translation pairs created will tend to be long and may not serve as good examples for example-based machine translation.

What we hope, therefore, is to allow, as far as possible, a sentence from one text to be matched with a sequence of clauses or phrases of the other text. But we first caution that it is impossible to pair up clauses of a Japanese sentence individually with clauses of a Chinese sentence as there will be cross referencing of different words or phrases among these clauses. However, if a period[*1] is used to end a sentence in one text, and even if there is no corresponding period but just a comma in the other text, that period serves to indicate a self-contained entity in the former text, and can be used to break up the sentence of the latter text at the location of the comma in question. But allowing matching of a sentence with clauses and phrases could lead to spurious matches with short clauses or phrases whose lengths may not be significant enough to give a conclusive match.

The above two problems, namely, the wider variability in the character ratio and the provision for matching with clauses and phrases, will reduce the accuracy of alignment. This is where we feel that common kanji appearing in both texts will provide a useful means to alleviate the problem.

3 Matching Strategy

Our matching strategy is to use periods in one text as clues to find break points in the other text, and vice versa. Thus there are two phases. The first phase is to break up Chinese texts into segments based on periods in the Japanese texts. The second phase will look for any period contained in each of the Chinese segments to break up the Japanese sentence with which that Chinese segment was matched in the first phase. We will use an example taken from [7] shown below to illustrate the two phases. Here, there are two sentences in each of the parallel texts. But the first period in both texts do not correspond to each other in position.

Japanese Text:
柔道の特長は「柔よく剛を制す」ということばに表れている。これは弱い者でも強い者に勝つことがあるという意味で、相手の力を上手に利用すれば、強い相手でも倒すことができるのだ。

Chinese Text:
柔道的特色表現在所謂的「以柔克剛」里，意味着弱者亦能戦勝強者。也就是説弱者若能善加利用強者的力量，一様可以撃倒対手。

The first phase will rely on the first period in the Japanese text to break up the Chinese text into matching segments as shown below:

(J) 柔道の特長は「柔よく剛を制す」ということばに表れている。

⟹(C) 柔道的特色表現在所謂的「以柔克剛」里，

(J) これは弱い者でも強い者に勝つことがあるという意味で、相手の力を上手に利用すれば、強い相手でも倒すことができるのだ。

⟹(C) 意味着弱者亦能戦勝強者。也就是説弱者若能善加利用強者的力量，一様可以撃倒対手。

In the second phase, it is noted that the second Chinese segment still contain a period which can be used a clue to break up the corresponding Japanese sentence, resulting in the final matching as follows:

(J) 柔道の特長は「柔よく剛を制す」ということばに表れている。

[*1] The same observation applies to other sentence punctuation marks such as question marks and exclamation marks, but we will treat them as periods for simplicity of discussion

$\Longrightarrow$(C) 柔道的特色表現在所謂的「以柔克剛」里，
(J) これは弱い者でも強い者に勝つことがあるという意味で、
$\Longrightarrow$(C) 意味着弱者亦能戦勝強者。
(J) 相手の力を上手に利用すれば、強い相手でも倒すことができるのだ。
$\Longrightarrow$(C) 也就是説弱者若能善加利用強者的力量，一様可以撃倒対手。

Dynamic programming technique is used to find the overall least cost in matching between sentences or clauses broken up in the above manner. In the absence of a simple probabilistic model as in [4], we have chosen a heuristic cost function to impose suitable constraints. The cost function is divided into two parts. The first part imposes constraints based on relations between the structures of the candidate pair. The second part looks for kanji as constraints to guide the match. We will call the first part S to stand for structural constraints and the second part K for kanji constraints.

There are three terms in part S. The first term computes the distance (i.e. the difference in lengths) between a sentence and its matched candidate as a measure of dismatch. In other word, the smaller the distance the better the match. The second is provided here because while we allow matching with clauses and phrases, we would still like to give an advantage to matched candidates that are full sentences as many sentences do match with full sentences. The second term therefore carries a negative sign to alleviate costs for such cases. With the third term, we would like to permit a few peculiar matching patterns but would associate each with a positive cost. There are basically three unusual patterns, namely 1-0, 0-1 and 2-x. The first two refer to matching involving an empty string while the last means two sentences being matched with x number of clauses or phrases. The last pattern is needed when there is no punctuation mark in the other text to find a suitable break point. For instance, the two Japanese sentences [7] "五月五日。この日は武者人形を飾り、こいのぼりを立て、ちまきやかしわもちを食べて祝います。" have to be matched with one Chinese sentence "五月五日這天是装飾武士人偶，樹鯉魚旗，吃粽子或柏餅来慶祝的日子。" as the first period of the Japanese sentence "五月五日。" cannot find a suitable break point in the first Chinese clause due to the absence of a comma at the appropriate place.

Part K in the heuristic function handles two kanji constraints. The first term is a measure of occurrence of common kanji found in both the sentence in question and its matched candidate. Bonus points are added if kanji found are consecutive in both texts. As the term actually provides a measure of correlation between the sentence and the matched candidate, it has a negative sign. The second term in part K has to do with an observation that in Chinese texts there are many two-character phrases such as "而且", "所以", "然而", "因此", and so on. Their shortness in length may not be significant enough to decide whether a break point is to be placed before or after the phrase. Fortunately, it is observed that such phrases will not trail behind a clause or a sentence. As such, a positive penalty cost is given to a matched candidate that has such a trailing phrase.

4 Alignment Algorithm

The two phases in the algorithm are very similar and can be described with the same set of notations. In the first phase, we call the Japanese text the *source* and the Chinese text the *target*, and vice-versa when the second phase is running. Recall that in each phase, the periods in the source are used to find break points in the source. Thus, in the source, the basic units are separated by periods and we call them *source units*. In the target, the basic units are the smallest units possible separated by any punctuation mark, and we call them *target units*.

We will first define a few notations. As we have either a source unit or a target unit, let w denote the type of a unit. Thus $w \in \{s,t\}$ where s and t denote source and target, respectively. Let C_z^w denote the number of characters in the zth unit of type w. Furthermore, let w_{xy} denote the length (in number of characters) from xth unit to yth unit of type w. While normally, $x \leq y$, we allow a special case where $x = y+1$ to mean an empty string. Thus we have for source length and target length, respectively, as follows:

$$s_{xy} = \begin{cases} 0 & \text{if } x = y+1 \\ \sum_{z=x}^{y} C_z^s & \text{if } 1 \leq x \leq y \end{cases} \tag{1}$$

$$t_{xy} = \begin{cases} 0 & \text{if } x = y+1 \\ \sum_{z=x}^{y} C_z^t & \text{if } 1 \leq x \leq y \end{cases} \tag{2}$$

Now let P_z^w denote the type of punctuation mark at the end of the zth unit of type w. Specifically we have:

$$P_z^w = \begin{cases} 1 & \text{if punctuation mark} \in \{\text{。 ！ ？ ；}\} \\ 0 & \text{if punctuation mark} \in \{\text{、 ， ：}\} \end{cases} \tag{3}$$

Let $align(i,u,j,v)$ be the heuristic function that computes the cost of aligning a sequence of ith to uth source units with a sequence of jth to vth target units.

As explained in the last section, $align(i,u,j,v)$ has two parts as follows:

$$align(i,u,j,v) = S(i,u,j,v) + K(i,u,j,v) \quad (4)$$

$S(i,u,j,v)$ computes the cost based on the structural relationship between the source and the target and contains three terms shown below as explained in the last section.

$$\begin{aligned} S(i,u,j,v) &= f_1|\rho s_{iu} - t_{jv}| - f_2(P^t_{j-1} + P^t_v) \\ &\quad + f_3(peculiar(i,u,j,v)) \quad (5) \end{aligned}$$

where ρ denotes the ratio representing the average number of target characters for each equivalent source character. The first term thus computes the distance measure between the equivalent source length ρs_{iu} and the target lenth t_{jv}. The second term examines the punctuation marks preceding the jth target unit and ending the vth target unit, respectively, and returns a cost appropriately. The last term looks for the presence of any peculiar matching of 1-0, 0-1 and 2-x, and adds a corresponding cost as follows:

$$peculiar(i,u,j,v) = \begin{cases} 1 & \text{if } j = v+1 \\ 1 & \text{if } i = u+1 \\ 3 & \text{if } i = u-1 \end{cases} \quad (6)$$

The values 1, 1 and 3 for 1-0, 0-1 and 2-x matching, respectively, represent their relative frequency of occurrence among the three patterns. Thus of all the three peculiar patterns found in our rough sampling, around 60% of them belong to 2-x, while the other two were about 20% each.

Next the kanji constraints are handled by part K as follows:

$$K(i,u,j,v) = -f_4(kanji(i,u,j,v)) + f_5(trailing(v)) \quad (7)$$

Here, the function $kanji(i,u,j,v)$ examines from the ith to uth source units to find the number of kanji that also appear in the jth to vth target units. For each pair of kanji found above that are consecutive in both the source and the target, a bonus count of 2 is also added. The number computed for common kanji is then divided by the total length of the above source units to give the average common kanji per source character. In fact, the occurrence of kanji is precomputed so that $kanji(i,u,j,v)$ is simply a table look-up during the alignment process. The second function $trailing(v)$ returns 1 if $t_{vv} = 2$ (i.e. the vth target unit has only two characters) and that the 2-character string $\in$ { 而且 , 所以 , 然而 , 因此 , ... }.

In quations 5 and 7, f_x $(1 \le x \le 5)$ are the weighting factors.

Now, let $match(u,v)$ represent the total cost of matching a string of source units from the beginning (i.e. 1st unit) to the uth unit with a string of target units from the beginning to the vth unit. By dynamic programming, the algorithm is such that the total cost $match(u,v)$ is minimum and may be expressed recursively as follows:

$$\begin{aligned} match(u,v) = \min_{i=\alpha}^{u-1} \min_{j=\beta(i)}^{\gamma(i,u,v)} & \{match(i-1,j-1) \\ & + align(i,u,j,v)\} \quad (8) \end{aligned}$$

where α, $\beta(i)$, and $\gamma(i,u,v)$ are defined as follows:

$$\alpha = \begin{cases} u+1 & \text{for phase 1} \\ u & \text{for phase 2} \end{cases} \quad (9)$$

$$\beta(i) = \begin{cases} v+1 & \text{if phase 1} \wedge i = u \\ v & \text{otherwise} \end{cases} \quad (10)$$

$$\begin{aligned} \gamma(i,u,v) = \min \; & \{ x \mid (1 \le x \le v) \wedge \\ & ((t_{xv} \le f_{max}\rho s_{iu} \wedge i \le u) \vee \\ & (t_{xv} \le L_{max} \wedge i = u+1))\} \quad (11) \end{aligned}$$

It is to be noted that the length of the source for matching may range from zero unit to at most 2 units, while the target length may range from zero unit to a number of units controlled by $\gamma(i,u,v)$ which gives an upper limit such that the target length will not be too long to be out of proportion for a reasonable match. α and $\beta(i)$ determine whether the source and target, respectively, will start with zero length, (i.e. $u+1$ and $v+1$, respectively). As all possible matches involving empty strings must have been captured in phase 1, there is no need to consider empty strings in phase 2. Thus α and $\beta(i)$ will be equal to $u+1$ and $v+1$ only in phase 1, and furthermore $\beta(i)$ will only be $v+1$ when the source has only one unit (i.e. $i = u$) because we will not allow 0-0 and 2-0 matching. $\gamma(i,u,v)$, as an upper limit, is to find an xth source target unit as far down from vth unit as possible such that the length t_{xv} will not be longer than the equivalent source length ρs_{iu} within a maximum allowance factor f_{max}. However, when the source is an empty string (i.e. when $i = u+1$), there is no source length to compare, in which case, t_{xv} will not be longer than a predetermined length L_{max}.

The recursion in equation 8 works backward and terminates when both texts have been perfectly aligned till their beginning or when one of the texts has been passed beyond its beginning (i.e. $u < 0$ or $v < 0$). As the latter case represents a mismatch, an ∞ cost is associated with it. Mathematically, the terminating condition is as follows:

$$match(u,v) = \begin{cases} 0 & \text{if } u = 0 \wedge v = 0 \\ \infty & \text{if } u < 0 \vee v < 0) \end{cases} \quad (12)$$

As mentioned in section 2, the average number of Japanese characters per Chinese characters was found to be 1.4 on average. Thus ρ is taken to be 0.7 (i.e. the reciprocal of 1.4) in phase 1 and 1.4 in phase 2. All other factors in the above equations were found by trial and errors through repeated testing with 20 bilingual passages. The following values were finally obtained: $f_1 = 1$, $f_2 = 10$, $f_3 = 2$, $f_4 = 80$, $f_5 = 10$, $f_{max} = 3$, and $L_{max} = 15$.

With all values in place, to align a Japanese texts containing n sentences with a Chinese texts containing m clauses or phrases, the program simply calls the routine $match(n, m)$. The two phases are then done automatically in succession.

5 Experimental Results

Altogether, we have tested the algorithm with 64 short bilingual passages taken from [7][8][9][10][11] [12][13][14] including the 20 passages that were initially used to determine the values of the factors in the algorithm. All the 64 passages contain a total of 270 Japanese sentences and 271 Chinese sentences. The fact that the total numbers of sentences in both texts are very close does not mean that one Japanese sentence will be mapped simply to one Chinese sentences in most cases. But rather some passages have more Japanese sentences than Chinese sentences while others vice-versa, such that some sentences have to be matched with clauses or phrases within sentences of the other text. If all the 64 bilingual passages are aligned correctly, it should produce 309 pairs of translation examples.

The accuracy of the algorithm is measured by dividing the number of correct pairs obtained from the algorithm by the total number 309. We have subjected the 64 passages to three different testing methods. The first was done by only allowing structural constraints to govern the alignment process, i.e. using only part S of the heuristic function in equation 4. The second was by only allowing kanji constraints (i.e. only using part K), and finally in the last method the whole function was used, namely, both part S and part K. The results are tabulated in Table 1.

Data Item		S	K	S+K
(1)	No. of passages aligned correctly	32	30	57
(2)	No. of correct pairs produced	216	230	294
(3)	No. of incorrect pairs produced	88	80	16
(4)	Percentage of accuracy rate	70%	74%	95%
(5)	No. of passages aligned correctly by any of the 3 methods	12	12	12
(6)	No. of passages aligned correctly only by 1 of the 3 methods	0	3	10
(7)	No. of passages aligned correctly by S+K and either S or K	20	15	35
(8)	No. of passages not aligned correctly by any of the 3 methods	4	4	4

表 1: Table 1: Experimental results for 3 different testing methods

Of the 64 passages, the numbers of passages aligned correctly by each of the three methods were 32, 30 and 57, respectively, indicating that both structural and kanji constraints are needed to produce good alignment results. However, those passages that were not aligned correctly in fact contained correct alignment pairs too among erroneous pairs. Therefore it would be more meaningful to see how many correct and incorrect alignment pairs were produced, which are shown in rows (2) and (3) of Table 1. It should be noted that the total numbers of correct and incorrect pairs produced did not necessarily add up to 309 as some errors were caused by merger or splitting of correct pairs. The accuracy rates for the three methods with respect to the 309 pairs, are 70%, 74% and 95%, respectively. It was indeed surprising to find that kanji constraints alone can achieve an accuracy of 74%, which is even slightly better than that by structural constraints only, namely, 70%. This appears to be in line with the observation that monolingual readers can manage a partial alignment by merely relying on common kanji in the bilingual texts.

Rows (5) to (7) in Table 1 give the following interesting observations. Of the 57 passages correctly aligned by the method S+K. 12 passages can be aligned by either S or K or both, meaning that either structural constraints or kanji constraints will do the job. Next, 10 passages can only be aligned correctly if both S and K are present, thus requiring both constraints together to provide the correct alignment. Then there were 35 passages that can be aligned by both S and K or by just either S or K alone. Of the 35 passages, 20 can be aligned correctly if both S and K are present or just S alone, indicating that S is the contributing component to the success of alignment. Similarly, the other 15 out of the 35 passages were aligned correctly with K being the contributing component, and whether S is present or not is immaterial. Examination of the contents of these passages shows that where structural patterns in the bilingual texts are quite regular (such as no unusually long or short translation), structural constraints alone would be sufficient. On the other hand, when common kanji are in abundance kanji constraints can become a dominating factor. Passages that have irregularities in translation but sufficient occurrence of common kanji thus still can be aligned correctly, as illustrated in the following example [7].

Japanese Text:

朝、上野から東北新幹線で岩手県の盛岡に行く。わずか3時間20分、東北新幹線が開通したのは1985年3月、以前は盛岡まで6時間かかったとか。私は開通したあと日本に来て運がよかったわ。

Chinese Text:

早上搭東北新干線从上野前往岩手県的盛岡。僅僅三小時二十分即抵達目的地。東北新干線是于一九八五年三月通車，据説以前到盛岡需花費六个小時，幸好我是通車后才来到日本。

If only structural constraints are used, phase 1 will produce the following intermediate result:

(J) 朝、上野から東北新幹線で岩手県の盛岡に行く。

⟹(C) 早上搭東北新干線从上野前往岩手県的盛岡。

(J) わずか3時間20分、東北新幹線が開通したのは1985年3月、以前は盛岡まで6時間かかったとか。

⟹(C) 僅僅三小時二十分即抵達目的地。東北新干線是于一九八五年三月通車，

(J) 私は開通したあと日本に来て運がよかったわ。

⟹(C) 据説以前到盛岡需花費六个小時，幸好我是通車后才来到日本。

A simple manual calculation of the total distance based on character counts between the matching bilingual units will show that the result indeed gives the best match over others but a visual inspection of the kanji contents will show its obvious error, which will lead to an incorrect final alignment in phase 2. The passage, however, has sufficient common kanji to give alignment cues. Thus by having both S and K or in fact just K alone will yield the following correct alignment in phase 1 and hence phase 2 as follows.

(J) 朝、上野から東北新幹線で岩手県の盛岡に行く。

⟹(C) 早上搭東北新干線从上野前往岩手県的盛岡。

(J) わずか3時間20分、

⟹(C) 僅僅三小時二十分即抵達目的地。

(J) 東北新幹線が開通したのは1985年3月、以前は盛岡まで6時間かかったとか。

⟹(C) 東北新干線是于一九八五年三月通車，据説以前到盛岡需花費六个小時，

(J) 私は開通したあと日本に来て運がよかったわ。

⟹(C) 幸好我是通車后才来到日本。

Row (8) in Table 1 shows that there were 4 passages which could not be aligned correctly by S or K or both. The errors were due to overwhelming structural irregularities over kanji constraints. Here again, these 4 passages are not entirely out of alignment as there are still correct pairs produced by the alignment algorithm. One final observation from Table 1 is in row (6) where there were 3 passages that could only be aligned by using kanji constraints alone but not both constraints. The structural constraints appear in this case to have detrimental effects when the constraints penalize irregularities in the texts.

6 Conclusion

We have presented an algorithm for automatic alignment of Japanese and Chinese parallel texts. While works on automatic alignment on other language pairs have been reported in the literature, our work here is unique in several respects. Firstly, the method handles two languages that are linguistically different while the earlier works [1][2][3][4] involve western languages with the exception of Hwang and Nagao's work[5] which deals with Korean and Japanese but the two languages have similar grammatical structures. Secondly, because of the sentential structure differences, we have decided not to do matching at the sentence level as in the earlier work but rather allow matching of sentences with clauses and phrases. Thirdly, the previous works used either lexical approach or statistical approach, the current work attempts to combine the two. Finally, unlike other lexical approaches which rely on some indirect means to obtain relationships between lexical contents of the bilingual texts, our work simply uses occurrence of common kanji to find a direct correspondence.

Because we allow matching with clauses and phrases, matching was more difficult and error-prone. The use of both structural and kanji constraints has, however, proven to be a remarkable means of guiding the automatic alignment with an accuracy rate up to 95%. We believe that the actual accuracy could be even higher as we tended to select more difficult passages for testing. Moreover, the factors used in the heuristic function may be further fine tuned to improve the accuracy. The current work, nevertheless, has shown that kanji constraints have indeed provided a unique and an interesting means to greatly improve the accuracy of automatic alignment.

One final remark about kanji: in addition to facilitating automatic alignment, we believe that there are further advantages to explore from the occurrence of common kanji. Once the sentences and clauses have been aligned from the Japanese-Chinese bilingual texts, finer correspondences may be possible within the confines of each matched pair. Kanji could here offer other possibilities in our future work. An interesting area that we hope to investigate is the construction of dependency trees from the matched pairs of sentences/clauses. The common kanji found in the matched pairs could then help in structural matching of these dependency trees as in the work by Matsumoto *et al.*[15] on the English-Japanese pair but there of course without the help of kanji. With matching of dependency trees, phrasal correspondences between Japanese and Chinese will then be realized for use in example-based machine translation as demonstrated in the work of Sato[16].

Acknowledgements

謝辞

Special thanks are due to Hitachi Scholarship Foundation for their support of this research.

References

参考文献

[1] R. Catizone, G. Russell and S. Warwick, Deriving translation data from bilingual texts, in *Proceedings of the First International Acquisition Workshop*, Detroit, Michigan, 1989.

[2] M. kay and M. Röscheisen, Text-translation alignment, Computational Linguistics, Vol.19, No.1, pp.121-142, 1993.

[3] P.F. Brown, J.C. Lai and L.M. Robert, Aligning sentences in parallel corpora, in *Proceedings of 29th Annual Meeting of the ACL*, pp.169-176, June 1991.

[4] W.A. Gale and K.W. Church, A program for aligning sentences in bilingual corpora, Computational Linguistics, Vol.19, No.1, pp.75-102, 1993.

[5] D. Hwang and M. Nagao, Aligning of Japanese and Korean texts by analogy, 類似性に基づいた日韓対訳テキストの文対応, 情報処理学会研究報告, 94-NL-99, Vol. 94, No. 9, pp.87-94, 1994.

[6] M. Nagao, A framework of a mechanical translation between Japanese and English by analogy principle, in *Artificial and Human Intelligence, eds. Elithorn and R. Baner*, Elevier Science Publishers, B.V., 1984.

[7] X.Y. Su (蘇秀玉), Latest events in Japan (最新日本事情), 致良出版社, 1992.

[8] W.Z. Liu (劉文柱) and R.S. Yu (于栄勝), Selected university Japanese articles, Vol. 1 (大学日語文選, 上冊), Beijing University Press (北京大学出版社), 1987.

[9] J. Chi (遲軍), Selected university Japanese articles, Vol. 2 (大学日語文選, 下冊), Beijing University Press (北京大学出版社), 1987.

[10] S.Y. Lan (藍三印), ed., A collection of NHK news articles (NHK 原音、日語新聞広播専集), 衆文図書公司, 1991.

[11] G.Q. Sun (孫国欽), Rapid technical Japanese (速成科技日語), 天津科学技術出版社, 1987.

[12] L.Q. Jing (靖立青), J.M. Ma (馬金水) and W.X. Liu (劉文祥), Bilingual scientific articles - "Force" (日語科普対照注釈読物 - "動力"), 商務印書館, 1981.

[13] M.Y. Gu (顧明耀) and D.D. Cai (蔡敦達), Selected scientific and technical Japanese articles on Computers (科技日語自学文選 - 電子計算机類), 商務印書館, 1984.

[14] Nihongo Journal (日本語ジャーナル), Japanese-Chinese edition, various issues from 1991.

[15] Y. Matsumoto, H. Ishimoto and T. Utsuro, Structural matching of parallel texts, in *Proceedings of the 32nd Annual Meeting of the ACL*, 1993.

[16] S. Sato, Example-based Machine Translation, PhD Thesis, Kyoto University, Sept 1991.