Inducing translations from officially published materials in Canadian government websites

Qibo Zhu
Statistics Canada &
Institute of Cognitive Science,
Carleton University
1125 Colonel By Drive,
Ottawa, Ontario,
Canada K1S 5B6
Qibo.Zhu@statcan.ca

Diana Inkpen
School of Information
Technology & Engineering,
University of Ottawa
800 King Edward,
Ottawa, Ontario,
Canada K1N 6N5
diana@site.uottawa.ca

Ash Asudeh
Institute of Cognitive Science &
School of Linguistics and
Language Studies,
Carleton University
1125 Colonel By Drive,
Ottawa, Ontario,
Canada K1S 5B6
ash_asudeh@carleton.ca

Abstract

Bitexts collected from web-based materials that are officially published on government websites can be used for a variety of purposes in language analysis and natural language processing. Mining officially published web pages can thus be an invaluable undertaking for translators in government departments who are producing the translations, and for machine translation researchers who are studying how those translations are produced. In this paper, we present the StatCan Daily Translation Extraction System (SDTES) and demonstrate how it is used to induce translations from officially published bilingual materials from government websites in Canada. New evaluation results show that SDTES is a very effective system for identifying and extracting sentences that are translation pairs from most of the federal government web pages which are currently under the CLF2 (Common Look and Feel for the Internet 2.0) framework.

1 Introduction

Well-aligned bilingual materials are a useful source of translations that can be used for translation studies, translation recycling, training data in machine learning, translation memories, information retrieval, machine translation, machine aided translation, and natural language processing (Jutras, 2000; Moore, 2002; Callison-Burch et al., 2005; Hutchins, 2005; Deng et al., 2006). In the past decade or so, an emerging trend in the use of bilingual texts has become rather noticeable: use of the web as a bilingual corpus (Resnik, 1999; Chen and Nie, 2000). With the increasingly widespread availability of vast quantities of web-based bilingual texts, more and more researchers are exploring ways to collect, from the web, bilingual texts of such language pairs as English-French, English-German, English-Italian, English-Chinese, English-Arabic and others (Kraaij et al., 2003; Resnik and Smith, 2003). Web-based materials that are officially-published bilingual materials on Canadian government websites contain exemplary translations, and mining these web-based bilingual materials can be beneficial to translators, linguists and researchers in many fields.

The StatCan Daily Translation Extraction System (SDTES) is a system that automatically extracts translations from the *Daily* news release texts of Statistics Canada (Zhu et al., 2007). In this paper, we will describe the algorithms and procedures of SDTES and present new results that show that SDTES can be effectively used to induce translations from other government websites in Canada. The paper is divided into 4 parts. The first part is this introduction. In the second part, we will analyze the general characteristics of the bilingual texts from Canadian government websites. The third part contains the algorithms and procedures we used to induce translations from web-based bilingual materials. In the fourth part, we will present evaluation metrics and results. The fifth part is the conclusion.

2 Features of web-based materials in Canadian government websites

Here are some general features that we observed of officially-published web materials on government websites in Canada.

1. It is required that federal government web pages in Canada be bilingual. For example, it is relatively rare for us to find a static HTML English page without a translated French page. Different departments or agencies may have different file naming conventions, but we generally do not have cases where two English pages correspond to one French page or vice versa.

2. Texts are normally translated by government employees who are trained in translation or text editing. There are rules and guidelines for them to observe about translating, editing and proofreading texts to be officially published. For example, without a good reason, editors are not allowed to add texts in one language to convey messages that are not present in the other language. This means that insertions and deletions at the sentence level will be very rare, if any.

3. Web pages have to be presented in the same way for the two languages. For example, if we use a heading (`h1`) for the title in the English page, we should do the same for the corresponding French page. Currently, almost all the federal government web sites are required to be under the CLF2 framework. Therefore we can expect to find many consistent correspondences in some main HTML markups such as `h1`, `h2`, `h3`, `table`, etc.

4. Important modules and templates of web pages normally go through usability tests before they are used for officially published translation products. One suggestion that often surfaces from users is that extra long web pages should be avoided when we can. If possible, long web pages should be either split into shorter pages, or rewritten to accommodate easy navigation and browsing. Therefore web pages on government websites are usually not very long.

5. Most texts are domain specific or text-genre specific, and use standard terminology of the specific department. The texts are more of a standard written style than of a spoken language style. Texts are mostly expository data and the translation style is typically not free translation. All these give rise to frequent translation correspondences at the word level, phrase level or sentence level. Some sites, such as the sites that publish data for statistics, banks, and finances, contain a lot of numerical data. Numbers can serve as important anchors for text alignment.

We used these observations as basic assumptions about the web pages of government websites in Canada, and designed algorithms and procedures to align the web pages and to detect potential errors in translation and in alignment.

3 Algorithms and procedures

SDTES contains two major components: one for bilingual text alignment and the other for misalignment detection. In this part, we present a set of protocols and algorithms that are designed for the two components.

3.1 Text mapping using the Gale-Church statistical model

The alignment component of SDTES is mostly based on the Gale and Church (1993) length model. The basic assumption of this model is that there is a strong likelihood that, for example, a long sentence in English will correspond to a long sentence in French; similarly a short sentence in one language will correspond to a short sentence in the other. Roughly speaking, if the average lengths of sentences in French and English are known, it is possible to set up a distribution of alignment possibilities from the sentence length information.

Suppose we have parallel texts S and T which can be split into n segments each. Each segment in S (s_i) is the translation of a segment in T (t_i), and they are aligned to form an alignment segment pair a_i. For each i, $1 \leq i \leq n$. A can be defined as the alignment of S and T that consists of a series of aligned segment pairs: $A \equiv \langle a_1, ..., a_n \rangle$. For the set B of all possible alignments $(A \in B)$, the goal is to find the maximum-likelihood alignment:

$$A_{\max} = \arg\max_{A \in B} \Pr(A \mid S, T) \qquad (1)$$

It is assumed that the probability of any aligned segment pair is independent of any other segment pair:

$$A_{\max} = \arg\max_{A \in B} \prod_{i=1}^{|B|} \Pr(a_i \mid s_i, t_i) \qquad (2)$$

The next assumption is that the length difference function $d(s_i, t_i)$ is the only feature influencing the alignment probability:

$$A_{\max} = \arg\max_{A \in B} \prod_{i=1}^{|B|} \Pr(a_i \mid d(s_i, t_i)) \quad (3)$$

By Bayes' theorem, $P(M \mid N) = \frac{P(N \mid M)P(M)}{P(N)}$, equation (3) becomes:

$$A_{\max} = \arg\max_{A \in B} \prod_{i=1}^{|B|} \frac{\Pr(d(s_i, t_i) \mid a_i) \Pr(a_i)}{\Pr(d(s_i, t_i))} \quad (4)$$

where the distributions $\Pr(d(s_i, t_i) \mid a_i)$ and $\Pr(a_i)$ can be estimated using hand-aligned data. When the normalizing constant $\Pr(d(s_i, t_i))$ is ignored and the logarithm is used,

$$A_{\max} = \arg\max_{A \in B} \sum_{i=1}^{|B|} \log \Pr(d(s_i, t_i) \mid a_i) \Pr(a_i) \; (5)$$

Gale and Church employed dynamic programming for establishing the optimal alignment to solve Equation (5).

When applying the Gale and Church algorithm in SDTES, we adopted a two-parse alignment procedure. Prior to the procedure, SDTES moves table contents and image contents to the end of the HTML page because in many cases these floating elements in HTML pages can end up in different paragraph positions in different languages. In some cases these elements can be the source of massive misalignment for the Gale-Church algorithm.

In the first parse, SDTES counts the few main HTML elements such as `h2, title,` and others to see if the pair of text files has the same number of main HTML features in them. If the numbers were the same, the system splits the texts into blocks separated by these feature HTML tags. SDTES first locates the paragraph boundaries in the texts and assign sentence delimiters to them. Then it assigns paragraph delimiters to text blocks that are divided by the main HTML tags. If the numbers of major HTML elements are different, the system treats the whole text document as a single text block, and assigns a paragraph delimiter to it. When the paragraph delimiters and the sentence delimiters are all assigned, SDTES does the first round of text alignment by the Gale-Church algorithm.

In the second parse, the system automatically reconstructs the English document and the French document from the paired paragraphs that are aligned in the first parse. SDTES locates the sentence boundaries in the aligned texts and assign sentence delimiters to them. Then SDTES separates the two halves in each aligned pair, and assigns a paragraph delimiter to each of them. After this, texts of different languages are collected and reassembled in two files, one for English and one for French. By reorganizing the text structures on the basis of the paragraph pairing results in the first around of text alignment, SDTES was able to reset the English and French documents to the original text format prior to the initial alignment. The two input files are processed with the newly assigned boundary symbols by the Gale-Church algorithm. The second parse produces text alignment at a more fine-grained text segment level.

3.2 Misalignment detection using anchor information

Anchor information includes positions or properties in one text which seem to match up with those in a parallel text. The information can be about structural features in the text or delimiters and markers that indicate “hard and soft boundaries” (Gale and Church, 1993:89), or “true points of correspondence” (Melamed, 1999:107) in alignment. Using anchor information, regions of text can be identified and alignments of text segments can be sought. Most alignment methods make use of anchor information in one way or another.

In STDES, we use the following structural features as anchors to assist misalignment identification.

HTML markups: Anchor information can be markups that go with the text and that reveal meta-information or style information about the text. For example some commonly used HTML tags such as `h1, h2, h3, br, p, hr, table, i, pre, form, img,` and `a` can be good anchors in bilingual text alignment. For a more detailed list of structural tags, format tags, content tags and irrelevant tags that can be used in alignment, see Sanchez-Villamil *et al.* (2006).

As indicated earlier, in Canadian government websites, we are likely to have a proliferation of such HTML markups. Usually, if a text in French contains a markup for a section in italics, then the corresponding section in English is likely to have the same markup. We did some HTML style unification formatting so that some parts of the HTML codes are highlighted, while some are ignored. For example, the code <a ...> becomes <a alink> after the unification formatting, because otherwise the link to an English webpage and the link to a French webpage might have been different.

Lexical units: Specific lexical units such as words or phrases can serve as anchor points (Brown et al., 1991). In SDTES, we mainly use cognate words as anchors to help locate traces of alignment deviations. Identification of cognates in SDTES is a two-step operation.

The first step is to produce candidate cognate lists using the K-vec algorithm (Fung and Church, 1994). The main objective is to find cognate candidates within an acceptable text-region range and limit the number of words to be considered as cognate pairs. The K-vec method was developed as a means of generating "a quick-and-dirty estimate of a bilingual lexicon" that "could be used as a starting point for a more detailed alignment algorithm …" (Fung and Church, 1994). We use the K-vec++ package (Pedersen and Varma, 2002) for the implementation of the K-vec algorithm. The package is called the `K-vec++` package because it extends the K-vec algorithm in a number of ways. Using the Perl programs in the `K-vec++` package, SDTES was able to obtain a very rough list that might contain cognates for each pair of documents, although there is a lot of noise in the list too.

In the second step, we apply an algorithm called Acceptable Matching Sequence (AMS) to identify true cognates from the candidate cognate lists generated by the K-vec algorithm. This is an algorithm that we designed and developed in SDTES. An AMS has two non-overlapping substrings that can be matched in the same order in both of the words in a cognate pair.

The algorithm extracts two substrings (θ^a and β^b) from a source word, say a French word (W_1), with a length threshold (T) for the two substrings combined. The purpose is to find if the target English word (W_2) has the two substrings θ^a and β^b in the same order in the string sequence.

Suppose $x = \min(length(W_1), length(W_2))$, and $y=\max(length(W_1), length(W_2))$, and z is the length difference threshold. $| length(W_1) - length(W_2)| \leq z$. We use the length difference threshold for initial filtering: if $y>10$ then $0 \leq z \leq 4$, otherwise $0 \leq z \leq 3$.

SDTES determines the T parameter with reference to y. Since T is the combined length of θ^a and β^b, $0\leq a\leq T$, $0\leq b\leq T$, $a+b=T$. The system discards word pairs with $y<4$. SDTES sets $T=8$ if $y>10$. For all the rest, SDTES uses the simple linear regression model $T=0.5y+1.8$ to compute the threshold value. This regression model is derived from the regression analysis of sample cognate pairs we picked from the StatCan *Daily* releases.

When matching θ^a and β^b in W_2, skipping some characters is acceptable before, after or between the two substrings; we call them "Don't Care Characters" (DCCs). The initial value a in θ^a is set to 0, and b in β^b to T. The system starts to match the two substrings θ^a and β^b in W_2. If a match is found for θ^a and β^b in W_2 in the right order, W_1 and W_2 are a cognate pair, if not, one substring θ^a is increased in length ($a=a+1$) while the other substring β^b gets decreased ($b=b-1$). The search continues till a two-substring match is found or $a>T$ or $b<0$.

The main advantage of using K-vec with AMS for cognate identification is that K-vec does not rely on sentence boundary information. In this way, it can help detect errors in alignment that depend on sentence boundaries. At the same time, AMS has the straightforwardness of the naïve matching algorithm of Simard et al. (1992), but avoids problems caused by common prefixes or by the requirement that the first four letters have to match. This improvement can increase the number of correct cognate pairs identified. At the same time, AMS inherits the strength of no-crossing-links constraint in the Longest Common Subsequence Ratio (LCSR) algorithm by Melamed (1999). Also, in limiting the number of substrings to be matched, AMS overcomes the inherent weakness of LCSR in positing non-intuitive links because of lack of context sensitivity, as noted in a recent study by Kondrak and Dorr (2004). This can help reduce the number of false positives for SDTES such as **cou**r**tiers**/**co**mp**uters**, m**ensu**e**ls**/r**esul**ts, and p**ar**u**ti**o**n**/st**artin**g. We found that the K-vec technique together with AMS algorithm is a good fit for cognate identification for the misalignment detection purpose in SDTES.

Numbers and punctuations: Similarly, numbers in texts can serve as anchors in alignment. They are good indications of correspondence because a number in one language is usually interpreted as a number in the other language. Some punctuation marks can also be anchor points. For example, if there is a question mark in English, normally we are expecting a corresponding question mark in its translation text in French.

When detecting misaligned portions of texts, what SDTES does is to parse every aligned text segment pair, and compares the features in each half of the pair before it arrives at one of the two decisions: *pass* and *problem*. The detecting process starts from two prior filtering mechanisms. One of them is the length ratio criteria. If a text segment in one language is more than 3 times longer than the corresponding text segment in the other language, the pair is marked as a problem pair. The second criterion is matching type. Because the extracted translations are independent translation pairs that will be used for translation memory systems and cross-language information retrieval systems, matching types like 1:0 and 0:1 have to be discarded. When these two criteria have been checked, SDTES compares the structural and lexical clues of the HTML text segments for further detection. These clues include selected cognates, punctuation marks, numbers and HTML tags. Finally, we have a no-match-tolerance principle: if there are no structural and textual clues present, and if the two prior filtering criteria (length and matching type) are checked, we mark the segment as *pass*.

When the misalignment detection process is completed, SDTES applies a filtering mechanism to the list of stored translations to eliminate the following types of aligned pairs: 1) Pairs that contain only meta-information coding or codes that are derived from the HTML coding unification process. 2) Aligned segments that include only the numerical information. 3) Duplicate sentences or similar constructions that have been seen more than once in the collection of texts. Sometimes this involves unifying or discarding some information such as numbers and tags in the text.

4 Results and performance evaluation

For evaluation, we measure the performance of the two main components of SDTES: one is the text aligning component and the other is the misalignment detection component. We used the files collected from http://www.forces.gc.ca for evaluation. These web pages contain 2,682 officially published news releases from April of 2000 to June of 2008. We assigned each page a file name beginning with a sequential number from 1 to 9. We grouped these files according to the initial number in the file name such as 1, 2, 3, …, 9, and randomly chose two files from each of the 9 groups. Then we manually aligned these 18 selected files to build the reference collection.

For the evaluation of the alignment component, we define M as the set of segments in the manually aligned reference collection, A as the set of machine aligned segments before the misalignment detection device is applied. Precision (P) and recall (R) are as follows:

$$P = \frac{|A \cap M|}{|A|} \qquad R = \frac{|A \cap M|}{|M|}$$

We used the same randomly selected files to evaluate the performance of the misalignment detection component. Here, the system proposed alignments (A') include those remaining pairs after the system has filtered what it identifies as misaligned segments. M is the reference set, i.e., the aligned translation units that the human evaluator thinks the system should have reported. The recall (R) represents the proportion of system-proposed translation segments (A') that are right with respect to the reference (M), and the precision (P) is the proportion of correctly proposed alignment segments with respect to the total of those proposed (A').

$$P = \frac{|A' \cap M|}{|A'|} \qquad R = \frac{|A' \cap M|}{|M|}$$

As we can see from Table 1, good results have been reported of the text alignment component. For the 18 aligned files used for evaluation, the overall alignment precision for the aligned translation pairs is 0.967; the recall is 0.976.

Table 1 also indicates that SDTES is accurate in detecting misaligned pairs (P=0.988 and R=0.974). A comparison of precision and recall for the data sets before and after the misalignment filtering shows the effect of the misalignment detection algorithm. Precision improved from 0.967 to 0.988 with a negligible loss of recall from 0.976 to 0.974.

	Text alignment		Misalignment detection	
File	P	R	P	R
1231	0.960	0.979	0.986	0.973
17	1.000	1.000	1.000	0.966
240	1.000	1.000	1.000	1.000
2653	0.875	0.933	0.929	0.867
388	1.000	1.000	1.000	1.000
372	1.000	1.000	1.000	1.000
452	0.981	0.987	0.987	0.981
413	1.000	1.000	1.000	1.000
532	0.980	0.980	1.000	0.980
57	1.000	1.000	1.000	1.000
661	0.850	0.895	1.000	0.947
681	0.840	0.875	0.917	0.917
719	1.000	1.000	1.000	0.971
784	0.903	0.929	0.971	0.957
891	0.824	0.848	0.967	0.879
872	1.000	1.000	1.000	1.000
957	0.956	0.956	0.955	0.933
982	1.000	1.000	1.000	1.000
Overall*	0.967	0.976	0.988	0.974

Table 1. Performance evaluation of the two main components in SDTES

While the SDTES model has achieved good results for officially published government bilingual text data on the web, there are also limitations. For the text alignment part, we still find some examples, although the number is very small, of chains of misaligned sentences. For about a dozen of so files in the collection, chain misalignment segments can be found where the Gale and Church algorithm finds it hard to come back to the right track. Although most of the misaligned pairs can be filtered by the misalignment detection mechanism, it is worthwhile to analyze these files to trace the causes of the misalignments. There are four types of texts that can trigger massive misalignment for the Gale and Church algorithm.

Alphabetical lists: In some government texts, there are lists of names, committees, or terms and definitions. Items in these lists usually land in different positions when sorted alphabetically in different languages. Because of the position changes, translations in these lists can hardly match on a line-to-line basis.

Swapped paragraphs: When paragraphs have rearranged positions because of blocks of texts such as “note to readers” sections or inserted menu items that are introduced by `div` and other HTML tags, the correct alignment chain can be disrupted.

Deletions and insertions: For any accidental deletions and insertions, once the system starts to align the wrong text segments, it usually continues to misalign a number of text segments before it can finally correct itself.

Alignment types: We also found a few examples where the existing 6 alignment types such as 0:1, 1:0, 1:1, 1:2, 2:1 or 2:2 were not enough to cover the right alignment. We need something like 3:1 or 3:2 to align them properly. When we give it an alternative alignment pattern, it would establish the wrong links with the following text segments and cause chain misalignment.

In assessing the performance of the misalignment detection device, we have to take into account two types of errors. One type involves some misaligned pairs that still go undetected. We call them *overlooked pairs*. The other type includes correct alignments that are wrongly labeled as misalignments and are thus mistakenly filtered in the process. We call these *overdone pairs*.

As can be seen from Table 1, the number of *overlooked pairs* that are not captured by the system was minimal: 98.8% of the identified correct alignment pairs are truly reliable translation pairs. Mostly, the *overlooked pairs* are pairs of partially correct alignments such as those close to pairs with 1:0 or 0:1 alignment patterns.

Generally speaking, *overdone pairs* are very few. When we examine these pairs, we found that inconsistent use of HTML codes and various ways of interpreting numbers were two notable sources of errors in misalignment detection.

Inconsistent use of HTML tags: In analyzing the results of the evaluation, we encountered some

[*] Computed over all the mistakes in all the documents.

examples in which HTML codes are not used consistently in the two halves of the aligned translation segments. For example, in some English sentences, we found the `span` tag, but the corresponding tag in French is `font`. In some cases, there are some HTML markups that are present in only one language, but not in the other. For example, we have some French sentences with the HTML tag `sup`, but the same tag is nowhere to be found in its corresponding translation segment in English.

Number interpretations: There are cases in which we have numbers on one side of the aligned pair, but not necessarily on the other side. In some cases, in English we find the number "10", but in French, the corresponding number is "X". In other cases, numbers are found in both texts, but the numbers are not the same. For example in Figure 1, using different numbers (1990 vs. 90) is not necessarily wrong, but it confuses SDTES and leads the system to misjudge correct translation pairs as *overdone pairs*.

e_1. In the early 1990s, despite larger declines in earnings in the North than in Canada, employment income remained higher.
f_1. Au début des années 90, malgré une baisse des gains plus prononcée dans le Nord que dans l'ensemble du Canada, le revenu d'emploi y est demeuré plus élevé.

Figure 1. Distinct numbers are used in the aligned translation pair

SDTES was initially built on the basis of the *Daily* releases of Statistics Canada published between 2004 and 2006. Then we extended the application to data collection of the *Daily* releases from 1995 to 2008. Altogether, we assembled 32,276 *Daily* release files (16,138 for each language). After filtering and formatting, SDTES generated 488,646 aligned text segments. In this study, we are using the model for five other government websites of which the Canadian Forces website is one, and more than 200,000 pairs of translations were generated. The aligned text segments are organized in an XML format for easy exportability into the translation memory system and other application systems. Meta information items about each of the aligned pairs were recorded, such as the length information (before the HTML codes are stripped), the source of the matched strings, and the matching patterns. Figure 2 includes a sample record of the final aligned segments. For the sake of presentation clarity, line breaks are added to different levels of XML elements.

```
<bead>
<en>It will ensure that our Canadian Forces in Afghanistan receive the supplies and equipment they need to get the job done.</en>
<fr>Il fera en sorte que les Forces canadiennes en Afghanistan reçoivent l'approvisionnement et le matériel dont elles ont besoin pour faire leur travail.</fr>
<pa>1:1</pa>
<id>2185:22</id>
<re>pas</re>
<le>120=150</le>
</bead>
```

Figure 2. SDTES XML output (modified version)

These aligned pairs of translations are clean and consistent, and most of them are free of translation errors. They are ready to be used in many applications and systems such as translation memory systems, information retrieval templates, and machine translation systems.

5 Conclusion

In this paper, we presented the algorithms and procedures of the StatCan Daily Translation Extraction System (SDTES) and demonstrated how it is used to induce translations from officially published materials in Canadian government websites. SDTES contains two main components: one for text alignment and the other for misalignment detection that we use to filter out possible translation errors. For the text alignment component, we used the Gale-Church algorithm for a two-parse alignment at the paragraph level and the text segment level. For misalignment detection, we used a few structural features as anchors for bitext comparison. The structural features include cognate words, numbers, HTML markups, punctuation marks, etc. In extracting cognates, we employed the K-vec algorithm in combination with our AMS algorithm. Our evaluation shows that SDTES has achieved good results and that it is a good model for inducing translations from officially-published materials on federal government websites in Canada beyond the Statistics Canada websites for which it was initially designed.

It would have been good if we could set the original Gale and Church algorithm (without the two-parse procedure) as the baseline, and compare it with the alignment method we adopted in SDTES to evaluate the effects of the two-parse procedure. However, this attempt was deterred by problems in identifying the paragraph boundaries without the help of HTML tags in the source files. If we run the algorithm without paragraph detection, the results would be very poor. We leave this to future work.

Acknowledgments

We thank Professor David Sankoff, Dr. Joel Martin, Professor Charles J. Fillmore, Professor Andrew Brook, and Professor Jim Davies for their invaluable input on various parts of the paper and for discussing the StatCan Daily Translation Extraction System with us. We also thank the three anonymous reviewers for their insightful comments.

References

Brown, P. F., J. C. Lai, and R. L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 169-176, Berkeley, CA.

Callison-Burch, C., C. Bannard, and Schroeder, J. 2005. A compact data structure for searchable translation memories. In *10th European Association for Machine Translation Conference: Building Applications of Machine Translation*, pages 59–65.

Chen, J. and J. Y. Nie. 2000. Parallel web text mining for cross-language IR. In *Proceedings of RIAO 2000: Content-Based Multimedia Information Access*, vol. 1, pages 62-78, Paris, France.

Deng, Y., S. Kumar, and W. Byrne. 2006. Segmentation and alignment of parallel text for statistical machine translation. *Natural Language Engineering*, 12(4):235-260.

Fung, P. and K. W. Church. 1994. K-vec: A new approach for aligning parallel texts. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 1,096-1,102, Kyoto, Japan.

Gale, W. A. and K. W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1): 75-102.

Hutchins, J. 2005. Towards a definition of example-based machine translation. In *MT Summit X. Workshop: Second Workshop on Example-Based Machine Translation*, pages 63-70, Phuket, Thailand.

Jutras, J-M. 2000. An automatic reviser: the TransCheck system. In *Proceedings of Applied Natural Language Processing*, pages 127-134, Seattle, WA.

Kondrak G. and B. Dorr. 2004. Identification of confusable drug names: a new approach and evaluation methodology. In *Proceedings of the 20th International Conference on Computational Linguistics*, vol. II, pages 952–958, Geneva, Switzerland.

Kraaij, W., J.-Y. Nie. and M. Simard. 2003. Embedding web-based statistical translation models in cross-language information retrieval. *Computational Linguistics*, 29(3): 381-419.

Melamed, I. Dan. 1999. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25(1): 107-130.

Moore, Robert C. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Machine Translation: From Research to Real Users, 5th Conference of the Association for Machine Translation in the Americas*, pages 135-144, Springer-Verlag, Berlin.

Pedersen, Ted and N. Varma. 2002. K-vec++: Approach for finding word correspondences. Available online: http://www.d.umn.edu/ ~tpederse/parallel.html.

Resnik, P. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Meeting of ACL*, pages 527-534, College Park, MD.

Resnik, P. and Noah A. Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29 (3): 349-380.

Sanchez-Villamil, E., S. Santos-Anton, S. Ortiz-Rojas and M. L. Forcada. 2006. Evaluation of alignment methods for HTML parallel text. In *Advances in Natural Language Processing, Proceedings of FinTAL 2006, 5th International Conference on Natural Language Processing*, pages 280–290, Turku, Finland, LNCS 4139. Springer, Berlin.

Simard, M., G. Foster and P. Isabelle. 1992. Using cognates to align sentences in bilingual corpora. In *Proceedings of the Fourth International Congress on Theoretical and Methodological Issues in Machine Translation*, pages 67-81, Montreal, Canada.

Zhu, Q., D. Inkpen and A. Asudeh. 2007. Automatic extraction of translations from web-based bilingual materials. *Machine Translation*, 21 (3): 139-163.