10-2014

Canvas: A fast and accurate geometric sentence alignment system using lexical cues within complex misalignment settings

Hussein M. Ghaly
Graduate Center, City University of New York

Canvas: A fast and accurate geometric sentence alignment system using lexical cues within complex misalignment settings

By

Hussein Ghaly

A master's thesis submitted to the Graduate Faculty in Linguistics in partial fulfillment of the requirements for the degree of Master of Arts, The City University of New York

2014

This manuscript has been read and accepted for the Graduate Faculty in Linguistics in satisfaction of the dissertation requirement for the degree of Master of Arts.

_____Andrew Rosenberg _____

_____	**_____**
Date	**Thesis Advisor**

_____Gita Martohardjono_____

_____	**_____**
Date	**Executive Officer**

THE CITY UNIVERSITY OF NEW YORK

Abstract

Canvas: A fast and accurate geometric sentence alignment system using lexical cues within complex misalignment settings

By

Hussein Ghaly

Adviser: Professor Andrew Rosenberg

In this paper, we present a new sentence alignment system (Canvas), which is a Python implementation of a geometric approach to sentence alignment, based on lexical cues. Canvas system is designed mainly to handle parallel texts exhibiting complex misalignment patterns, namely within English-Arabic pairs for United Nations documents. The system relies heavily on pre-indexing words/tokens in the source and target texts, and it creates correspondences between the token indexes. From this point onward, the alignment problem is reduced to a geometric problem of finding the path that runs through the True Correspondence Points (TCPs). The likelihood of a point being a TCP depends on the clustering of other points nearby; so, we collect the most likely points, and we identify the shortest path containing the maximum number of these points using a modified form of Dijkstra's algorithm. The results of Canvas system are very promising, as they demonstrate that it can handle intricate misalignment patterns, with much better speed than other alignment approaches using lexical cues, and with good accuracy in general, in a completely automated fashion. The only drawback is that the system does not cover all the alignment segments and this coverage is generally lower than other systems, which can be a subject of future research.

Acknowledgements

I would like to thank my supervisor, Dr. Andrew Rosenberg, for his continuous guidance and support throughout the alignment project. I am also thankful for Dr. Matt Huenerfauth who has provided a lot of help for the project during the early stages. I would also like to thank Dr. Liang Huang for providing me with evaluation material. From the United Nations, I would like to thank my manager, Ms. Latifa Amine Saint-Roche, who has provided much-needed support for the alignment project at the early stages. Also Ms. Cecilia Elizalde has provided me with a lot of guidance and connections for taking the project further, for which I am really grateful. My colleagues Michal Ziemski and Jose Garcia-Verdugo (Pepe) have provided me with good ideas, suggestions and feedback at several points. I would also like to thank Mr. Bruno Pouliquen and Mr. Christophe Mazenc for their explanation of many points related to alignment during their sessions on Machine Translation.

I would finally like to express my deep gratitude for my wife Randa and my children Salma and Yaseen for everything they have given me, and for bearing with me during all the difficult times.

Table of Contents

Lists of Tables

List of Figures

Introduction

Alignment of parallel/bilingual corpora on the sentence level is a topic of great relevance to a variety of domains and applications. Sentence alignment (or bitext/parallel text alignment) is a process of mapping sentences in the source text to their corresponding units in the translated text (Li et al, 2010). It is very expensive and time consuming to manually align such corpora, where they can span into hundreds of thousands of documents, so there is a need for automatic systems to produce well-aligned parallel corpora.

Machine translation is one of the major applications that depend on sentence alignment, where this task is the first stage in extracting structural information and statistical parameters from bilingual corpora (Wu, 1994). Sentence alignment is also a prerequisite for word alignment (Gale and Church, 1991). Ultimately, in order to apply machine learning to machine translation, sentence-aligned parallel bilingual corpora have proved very useful (Moore, 2002).

Other tasks in computational linguistics depend on sentence alignment as well. Such tasks include cross language information retrieval, word sense disambiguation (Ma, 2006), statistical Natural Language Processing, algorithms based on unsupervised learning, automatic creation of resources (Singh and Husain, 2005), automatic extraction of translation equivalents, automatic creation of concordances (Gomes, 2009), multilingual categorization, training and testing multilingual information extraction software, automatic translation consistency checking, training of multilingual subject

domain classifiers (Steinberger et al, 2006). Other tasks, such as multilingual and monolingual lexicography, computer-aided language learning and translation studies, also depend on the availability of aligned parallel corpora (Tomeh, 2012).

Aligned parallel corpora are essential in the contrastive study of the language in general, and they provide material to gain more insight into cross-linguistic phenomena and processes. Many researchers have used such corpora to investigate sentence structure and word order in different languages, such as (DeNero and Uszkoreit, 2011). Other efforts, such as The Parallel Grammar Project, have used parallel corpora in multiple languages, to test the universality of Lexical-Functional Grammar (LFG) formalism, which assumes a version of Chomsky's Universal Grammar hypothesis, namely that all languages are structured by similar underlying principles (Butt et al., 2002).

In the domain of Second Language Acquisition, researchers have started to use parallel corpora for second language research and teaching. This is because such corpora provide the basis for a more accurate and reliable description of how languages are structured and used rather than based on perceptions and intuitions. Therefore, such parallel corpora allow language learners to compare contexts and become more aware of different uses of words in different contexts, and so they will be more able to see subtle differences between the native and the target language (Tsai and Choi, 2005). Aligned parallel corpora are also quite essential in Second Language Acquisition, as they can be used to predict and diagnose the performance of language learners, as was the case for Chinese

learners of English in their use of English passives, where the learner corpus is contrasted with the parallel corpus (Xiao, 2007).

In Human Translation, having aligned parallel corpora is very essential, as it provides reference translation that can be looked up easily, which facilitates the translation process considerably. The case is also the same for Computer Aided Translation tools, such as TRADOS, which can be equipped with a memory-based module that can find the translation from a large database of exact or similar matches from sentences or phrases that are already known from the parallel corpora (Khadivi, 2008).

There have been several approaches to alignment, which are discussed in this paper; however, there was generally a lack of evaluation material (gold standard annotated parallel documents), to measure the accuracy of each alignment approach and how well it performs for different situations. Therefore, one of the main contributions of this project was to create a nucleus for such evaluation data with a few annotated parallel documents in this language pair and in this domain to help future research on this subject.

The first part of the paper is organized in such a way to provide:

- Categorization of patterns of misalignment that any sentence alignment approach should consider
- Description of general challenges to alignment, and specific challenges related to the language pair (in our case English-Arabic) and the domain (in our case United Nations documents).

- Description of the data used in the alignment process.
- Analysis of different alignment approaches and their strengths and weaknesses.

The second part introduces

- The approach used, and how it differs from previous approaches
- Experimental setup and the evaluation criteria
- Analysis of the results achieved.

Alignment Challenges

The main challenge for alignment is that sentences/segments do not necessarily map one-to-one, and there are many possible patterns for misalignment, as will be shown in the following section. The bottom line in the alignment process is to identify certain cues, from which it would be possible to tell which segments align to which.

Among the most obvious cues are the sentence lengths (the number of characters or words in the sentence); where shorter source sentences align to shorter target sentences and longer source sentences to longer target ones. However, some factors may inhibit the effectiveness of the length criteria; e.g. consecutive sentences with similar lengths, inconsistent length distributions (such as when expanding an acronym). Also, it can also be the case that source sentences and target sentences follow the same length distribution but they are not actually translation of one another (as in the case for alphabetical ordering of each set of segments).

Alternatively, using lexical cues can help provide more confidence about the sentences being more likely translations of one another. The major drawback cited by almost all lexical-based approaches is that there are heavy processing requirements and the alignment process is generally much slower than the length based approaches. The problem also with lexical cues is that they are not always available, where they should be available in machine readable format (Machine Readable Bilingual Dictionaries) (Melamed, 1996). Even with the availability of such dictionaries/lexicons, it may often be the case that the words in the source segments are context sensitive or are within idiomatic expressions so their typical corresponding words will either be absent from the correct target segment or they would map to wrong segments. This is in addition to the typical problem that there will be many consecutive sentences which do not have words within the lexicon, so they may be described as “text deserts”, where there are no cues to know which segment map to which.

In addition to the above mentioned challenges, there are more specific challenges within English-Arabic pairs, and also with United Nations documents.

Arabic-Specific Challenges

1- Arabic morphology: The affixation system in Arabic is not straightforward, as we can see from the example in table 1, where one English word can correspond to many Arabic tokens, which are essentially various forms of the same word.

Table 1 – Complexity of Arabic Morphology

Report	Taqrir/تقرير
Submitted a report	Taqriran/تقريرا
His report	Taqrirahu/تقريره
Her report	Tarqiraha/تقريرها
And their report	wTaqrirahum/وتقريرهم
In my report	bTaqriri/بتقريري
And to our report	wlTaqrirana/ولتقريرنا
The report	alTaqrir/التقرير

So, if the word pair in our lexicon is Report:Taqrir/تقرير , we will not be able to match the other forms. So, the challenge is to be able to systematically stem any word consistently to its base form.

2- Arabic Orthography: Some lexical-based approaches rely heavily on cognates to substitute/complement the use of lexical cues; however, this applies mainly to similar language pairs; e.g. English-French, but not to languages with completely different orthography, such as English-Arabic.

3- Arabic word order: Some geometric approaches assume the correct alignment would have the words/tokens in the most linear fashion. However, this is not the case in English-Arabic pairs, on two counts at least:

- English sentences follow the Subject-Verb-Object (SVO) order, while Arabic sentences typically follow Verb-Subject-Object (VSO) order.

- English Adjective Phrases are the exact opposite order of the corresponding Arabic Phrase, as in the below example (notice that Arabic text goes from right to left):

(1) General (2) Temporary (3) Assistance (3) المساعدة (2) المؤقتة (1) العامة

Table 2 – Word order Difference between English and Arabic

General (1)	Temporary (2)	Assistance (3)
(3) المساعدة	(2) المؤقتة	(1) العامة

4- Arabic length considerations: While typically Arabic sentences are shorter than English sentences, there maybe certain situations where the Arabic sentence is considerably longer, as shown below:

Table 3 – Cases where English sentences are longer than Arabic sentences

Case	English Phrase	Arabic Equivalent
For certain new terminology	Gender Mainstreaming	تعميم مراعاة المنظور الجنساني
For acronyms	UNDP	برنامج الأمم المتحدة الإنمائي

United Nations Specific Challenges

There are many editorial considerations within the United Nations documents that cause and exacerbate the problem of misalignment, for example:

1- Alphabetical listing: Countries (and other entities) are typically sorted according to their alphabetical order. This means that their order in each language is different.

2- Sections displacement: this can also be dependent on the alphabetical sorting of the section header. Some alignment approaches assume that segments IDs are continuously increasing, while in situations like these it can be the case that as we progress with the source segments and find increasing target segments we may encounter a new section that is at earlier part of the document and hence has lower target segments IDs, which can be shown in the negative offset pattern, a misalignment patterns in the following section.

3- Untranslated text: This is mainly in footnotes and end notes, and it can be useful to identify segments which simply contain the same words. However, the challenge it poses is that any of such text segments can be a best match to many source segments which do not have enough lexical cues, so these untranslated segments need to be aligned first somehow.

Misalignment Patterns

A parallel text (from now on it will be called bitext) is extracted from two texts, and there is no guarantee that the text segments from each text are aligned together. In fact, there are many factors that cause and exacerbate the bitext misalignment, including the following:

1- Differences in formatting

2- Differences in segmentation rules (how text is split into sentences/segments)

3- Mistakes and omissions/additions/changes of some punctuation

4- Translation style, for example some segments and sections in United Nations documents are sorted according to the alphabetical order in each language for the source text and the target text

The following figures indicate the bitext maps for the possible misalignments within any parallel text, based on ad-hoc inspection. A bitext map is a list of correct pairs of segment IDs in each text, where the x-coordinate is the source segment ID (English in this case) and the y-coordinate is the target segment ID (Arabic in this case). The correct alignment would be if each segment is aligned to a segment with the same ID; hence the bitext map would coincide with the diagonal, as shown in figure 1 below. The following figures indicate other modes involving some misalignment, and it is likely that a misaligned document would contain a combination of these modes, in addition to the possibility of omissions/deletions or additions.

It should be noted that in this paper we are aligning English-Arabic bitexts, but in the below examples, we use English-French only to be able to visually identify each pattern of misalignment, due to the similarity between these two languages.

1- Correct Alignment

Figure 1 – Correct Alignment Segment Correspondence Graph

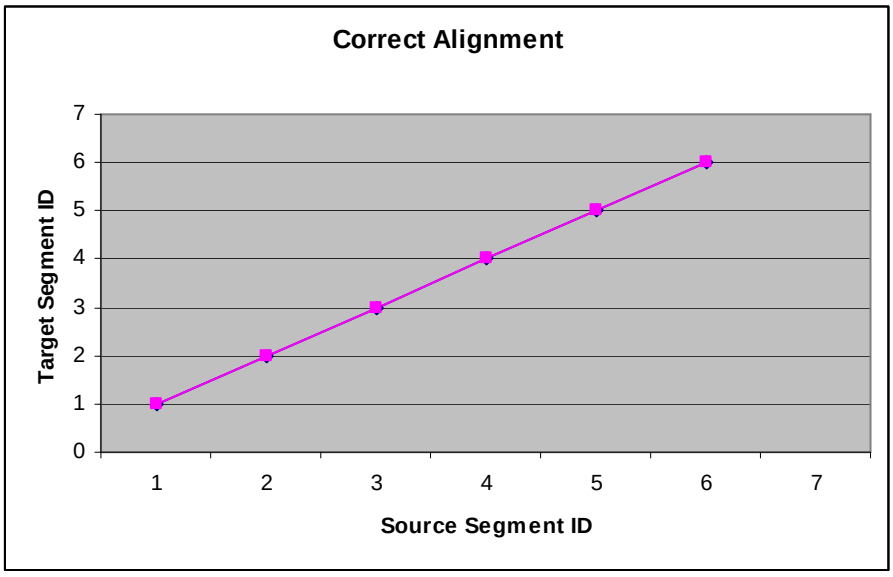

Figure 2 – Correct Alignment Segment Correspondence Bitext

Source Segment ID	Source Segment	Target Segment ID	Target Segment
1	United Nations	1	Nations Unies
2	CEDAW/C/SR.992	2	CEDAW/C/SR.992
3	Convention on the Elimination of All Forms of Discrimination against Women	3	Convention sur l'élimination de toutes les formes de discrimination à l'égard des femmes
4	Distr.: General	4	Distr. Générale
5	Chair: Ms. Ameline (Vice-Chair)	5	Présidente : Mme Ameline (Vice-Présidente)
6	In the absence of Ms. Pimentel, Ms. Ameline, Vice-Chair, took the Chair.	6	En l'absence de Mme Pimentel, Mme Ameline, Vice-Présidente, assume la Présidence.

This mode simply indicates that the text segments extracted from the two documents are in the same order and they are matching each other. It is possible to do some careful effort at the early stage of extracting text from documents to make sure that the segments are as close as possible to this mode, as it can save more effort in the alignment stage later on, but at any rate, it is unavoidable to have some misalignments within any bitext.

2- Positive Offset

Figure 3 – Positive Offset Segment Correspondence Graph

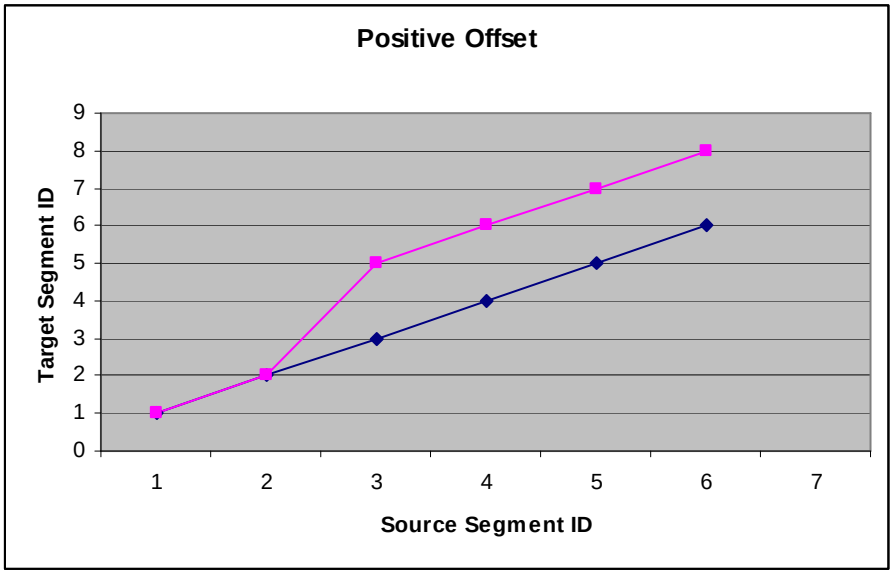

Figure 4 – Positive Offset Segment Correspondence Bitext

Source Segment ID	Source Segment	Target Segment ID	Target Segment
1	United Nations	1	Nations Unies
2	CEDAW/C/SR.992	2	CEDAW/C/SR.992
3	Convention on the Elimination of All Forms of Discrimination against Women	3	(
4	Distr.: General	4	A
5	Chair: Ms. Ameline (Vice-Chair)	5	Convention sur l'élimination de toutes les formes de discrimination à l'égard des femmes
6	In the absence of Ms. Pimentel, Ms. Ameline, Vice-Chair, took the Chair.	6	Distr. Générale
7	{	7	Présidente : Mme Ameline (Vice-Présidente)
8	-	8	En l'absence de Mme Pimentel,

			Mme Ameline, Vice-Présidente, assume la Présidence.

This mode indicates that there have been some spurious segments that led to a shift in the order of segments, creating a positive offset, where the target segments are above the diagonal. These spurious segments can be additions in the target language or they can be segments corresponding to other source segments somewhere in the document.

3- Negative Offset

Figure 5 – Negative Offset Segment Correspondence Graph

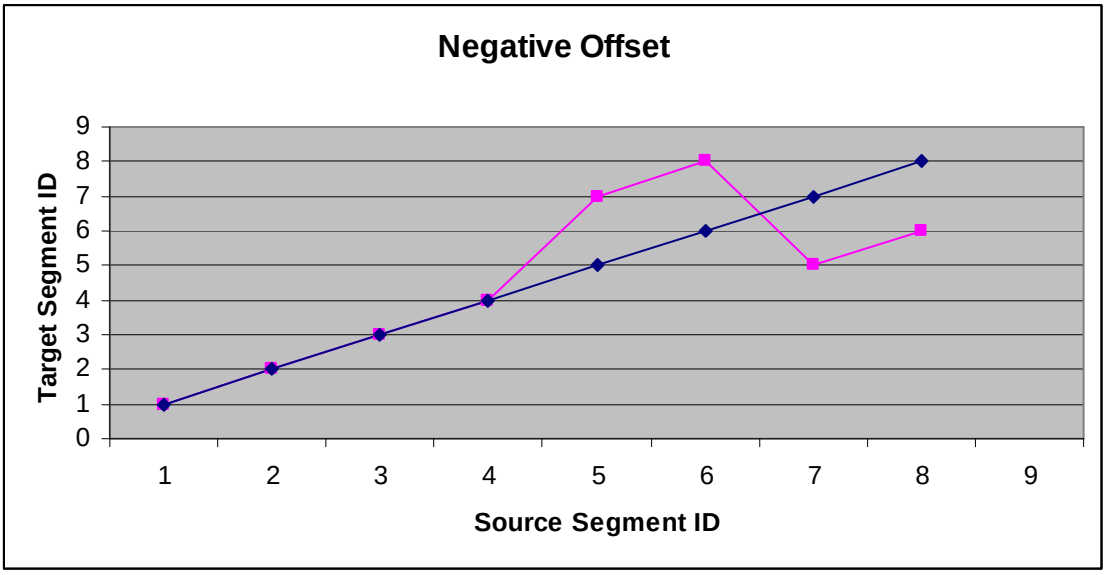

Figure 6 – Negative Offset Segment Correspondence Bitext

1	United Nations	1	Nations Unies
2	CEDAW/C/SR.992	2	CEDAW/C/SR.992
3	Convention on the Elimination of All Forms of Discrimination against Women	3	Convention sur l'élimination de toutes les formes de discrimination à l'égard des femmes
4	Distr.: General	4	Distr. Générale
5	A	5	Présidente : Mme Ameline (Vice-Présidente)
6	B	6	En l'absence de Mme Pimentel, Mme Ameline, Vice-Présidente, assume la Présidence.
7	Chair: Ms. Ameline (Vice-Chair)	7	A
8	In the absence of Ms. Pimentel, Ms. Ameline, Vice-Chair, took the Chair.	8	B

This negative offset may occur due to spurious segments on the target side, but it may also occur in the cases where the sections of the documents have different order in the source and in the target document (for example if the sections are alphabetically ordered, so each document will have a different order). This particular alignment pattern is very tricky, because some alignment approaches assume that the order of both the source and target segments IDs are always ascending, so if the segment order is returning to an earlier part of the document, probably they would be considered as deletions from one side and additions to the other, without being correctly aligned to each other.

4- Spiked Misalignment

Figure 7 – Spiked Misalignment Segment Correspondence Graph

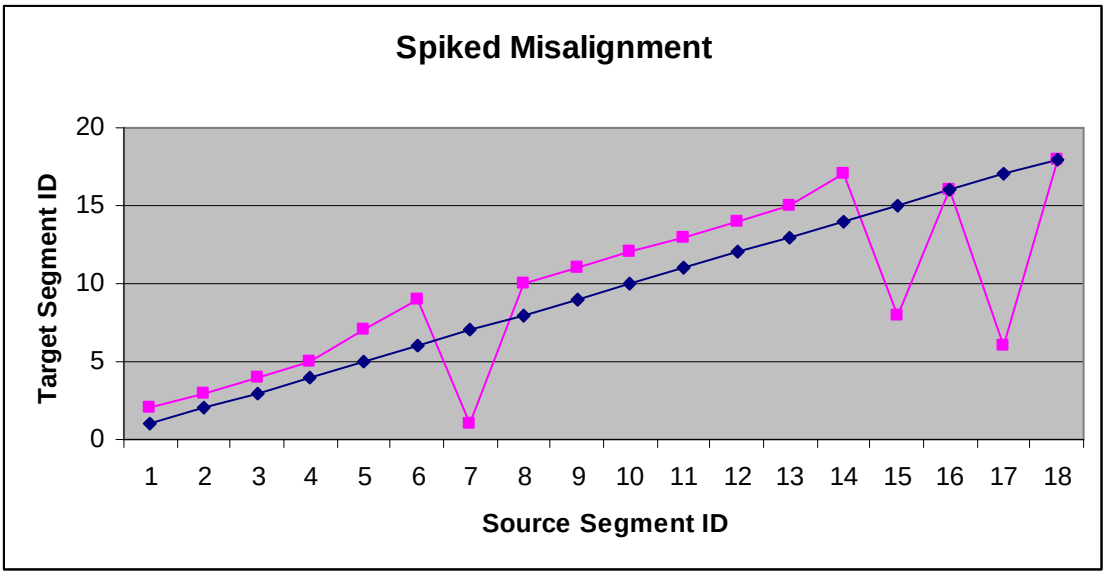

Figure 8 – Spiked Misalignment Segment Correspondence Bitext

1	Austria	1	Allemagne
2	China	2	Autriche
3	Congo	3	Chine
4	Côte d'Ivoire	4	Congo
5	Ethiopia	5	Côte d'Ivoire
6	France	6	États-Unis d'Amérique
7	Germany	7	Éthiopie
8	Japan	8	Fédération de Russie
9	Libya	9	France
10	Namibia	10	Japon
11	Nigeria	11	Libye
12	Panama	12	Namibie
13	Philippines	13	Nigéria
14	Republic of Moldova	14	Panama
15	Russian Federation	15	Philippines
16	Syrian Arab Republic	16	République arabe syrienne

17	United States of America	17	République de Moldova
18	Uruguay	18	Uruguay

This spiked misalignment occurs typically within tables containing alphabetically labeled items. It can also occur due to problems with document formatting, but usually aligned segments cluster together, so it is unlikely to have the correct segments dispersed around the document, which gives some advantage to geometric alignment approaches, because we will not be considering the alignment of each isolated segment, but of the segment within its neighboring segments.

5- One to Many Misalignment

Figure 9 – One to Many Misalignment Segment Correspondence Bitext

1	United Nations CEDAW/C/SR.992 Convention on the Elimination of All Forms of Discrimination against Women Distr.: General	1	Nations Unies
		2	CEDAW/C/SR.992
		3	Convention sur l'élimination de toutes les formes de discrimination à l'égard des femmes
		4	Distr. Générale
2	Chair: Ms. Ameline (Vice-Chair)	5	Présidente : Mme Ameline (Vice-Présidente)
3	In the absence of Ms. Pimentel, Ms. Ameline, Vice-Chair, took the Chair.	6	En l'absence de Mme Pimentel, Mme Ameline, Vice-Présidente, assume la Présidence.

This pattern usually occurs when there is discrepancy between the segmentation rules between the source and the target, or simply because there were some line breaks in one document and not in the other.

6- Many to One Misalignment

Figure 10 – Many to One Misalignment Segment Correspondence Bitext

1	United Nations	1	Nations Unies CEDAW/C/SR.992 Convention sur l'élimination de toutes les formes de discrimination à l'égard des femmes
2	CEDAW/C/SR.992		
3	Convention on the Elimination of All Forms of Discrimination against Women		
4	Distr.: General	2	Distr. Générale
5	Chair: Ms. Ameline (Vice-Chair)	3	Présidente : Mme Ameline (Vice-Présidente)
6	In the absence of Ms. Pimentel, Ms. Ameline, Vice-Chair, took the Chair.	4	En l'absence de Mme Pimentel, Mme Ameline, Vice-Présidente, assume la Présidence.

This pattern is the same as the previous one, except for the fact that the source and target are swapped.

Data

File Types

Our alignment system Canvas is designed mainly to handle bitext documents contained in html tables, so our experimental data are all are all in this file format. However, it can equally handle documents that are in text files, where segments are lines within these files. The system can also handle MS word documents in terms of extracting text from source and target documents and feeding both texts to the alignment system pipeline, but it was not tested to make sure it performs reliably on such documents, which usually involve many complex elements. This can be a subject of future research to investigate if improving text extraction (from MS Word files mainly) and text segmentation would make subsequent alignment task more accurate, and to find ways to achieve this improvement.

Lexicon

Since our approach is based mainly on lexical cues, we need a lexicon of word pairs, which is used during the alignment process. The lexicon can be prepared manually within CSV or XLS file and updated with new word pairs as necessary. However, in order to generate as many word pairs automatically, we singled out a collection of reasonably aligned bitexts, and proceeded as follows:

1- Index all the words in the source and target segments within each document
2- Create inverted index for each word in the source and target
3- For each source word, identify a sample of the segments where it occurs, and the words in the corresponding target segments
4- Identify the most corresponding target word as follows:

$$Word_{target} = argmax_{w_t} \frac{len(indexes(word_{source}) \bigcap (indexes(w_t))}{len(indexes(word_{source}) \bigcup (indexes(w_t))}$$

5- Collect the word pairs, and prune word pairs which have one word in common by choosing the one with the highest correspondence ratio

It should be noted that the lexicon needs to be as accurate as possible, since noisy word pairs may affect the alignment process at later stages. However, this concern should be weighed against the effect of word pair scarcity, which may leave many segments without lexical cues.

Related Work

The major milestone for sentence alignment was when (Gale and Church, 1991) developed their alignment algorithm, which assigns a probabilistic score to each proposed correspondence of sentences, based on the scaled difference of lengths of each two sentences (in characters) and the variance of this difference. Dynamic programming is used with this probabilistic score to find the maximum likelihood alignment of sentences.

Although this approach is very simple, it was quite successful; however, this algorithm and subsequent length-based algorithms are not robust with respect to non-literal translations and deletions because they ignore word identities within segments of similar lengths (Chen, 1993).

Subsequent length-based approaches (e.g. Brown et al. , 1993) tried to use the sentence length in words and assign anchor points to improve the alignment, however, the process of creating these pivots involved manual work, which may contradict the point of automatic alignment.

Other subsequent approaches tried to depend increasingly on lexical cues, as (Wu, 1994), who tried to adapt Gale and Church length-based algorithm to English-Chinese pairs, while integrating a set of words with invariant translation as pivots within the document. Also to maximize the value of lexical cues, it was suggested by (Kay and Roscheisen, 1993) to rely mainly on content words to avoid the noise created by other words, and to use the distribution of these words as a cue to the alignment.
Also using lexical cues, (Chen, 1993) devised an algorithm to identify the probability of one segment corresponding to another by the probability of the sentence beads (groups of words) between the two segments.

One of the most important tools in the lexical-based alignment is Champollion, which was developed by (Ma, 2006). This system uses mainly lexical cues, and calculates the similarity between segments using dynamic programming. Lexical cues are weighted

according to their Term Frequency- Inverse Document Frequency (TF-IDF). Length difference is integrated in this approach as a penalty.

However, most of these approaches are based on some paradigm that gives much significance to the segment boundaries, although this is not necessarily true, as segments in many cases are arbitrary, and it has been encountered that one source segment can map to even more than 10 target segments and vice versa.

In contrast, the alignment approach advocated by (Melamed, 1996) is based on the idea that we need to align words/tokens together, rather than segments. Tokens are identified by their distance (in characters) from the origin (beginning of the document). Source tokens are on the x-axis and target tokens are on the y-axis, and if a source token and a target tokens are matching (mainly being cognates), they are depicted as a point on the coordinate system, where the x-coordinate is the distance of the source token from the origin and the y-coordinate is the distance of the target token from the origin. This step is further clarified in our Approach section.

So, Melamed's approach starts with a rectangular window starting from the origin, and examining the points enclosed by the rectangle, whether they represent correct pairs, or True Correspondence Points. The points are organized into chains according to their distance from the diagonal. The correct chains, which have the TCPs are characterized with:

1- Linearity: points tend to line up straight.
2- Constant slope: the slope of the chain is similar to the slope of the diagonal of the bitext.
3- Injectivity: no two points in the chain have the same x-coordinate or y-coordinate.

The rectangle is expanded till a chain that satisfies this criteria is found, then a new rectangle is formed from the maximum point of the chain.

This approach does not, however, handle many of the challenges of the language pairs involving Arabic, especially where it comes to linearity. Also an approach involving Arabic would need to have its own lexicon to match tokens together because cognates cannot be used. Also, this approach cannot handle certain misalignment patterns (e.g. negative offset and spiked misalignment).

Approach

The alignment approach introduced here consists of the following stages:

1- **Initialization:**

In this stage, the source segments (English) and target segments (Arabic) are loaded, removing unnecessary characters and HTML tags, while loading the list of word pairs (lexicon) as well. The segments are then tokenized.

2- **Forward Indexing:**

This stage is based mainly on Melamed's approach described in the related work section. We start with the following example segments:

Table 4 – Example of source segments in English

Segment ID	Segment text
0	United Nations
1	Financial report and audited financial statements
2	for the biennium ended 31 December 2009
3	and Report of the Board of Auditors

Table 5 – Example of target segments in Arabic

Segment ID	Segment text
0	الأمم المتحدة
1	التقرير المالي والبيانات المالية المراجعة
2	عن فترة السنتين المنتهية في 31 كانون الأول/ديسمبر 2009
3	وتقرير مجلس مراجعي الحسابات

After tokenizing each segment, we identify the following information for each token, contained in the following index element:

Segment ID to where token belongs	Token location (in characters from the start of the first segment)	Token	Token number (within the forward index)

This index element gives a clear and unique identification for each token, which is going to be very helpful in later stages. These index elements are combined into a forward index for all the source and target segments, as follows:

Table 6 – Example of Source Forward Index

Segment ID	Location	Token	Number
0	3	United	0
0	9	Nations	1
1	18	Financial	2
1	26	Report	3
1	35	Audited	4
1	43	Financial	5
1	53	Statements	6
2	73	Biennium	7
2	79	Ended	8
2	83	31	9
2	88	December	10
2	94	2009	11
3	108	Report	12
3	118	Board	13
3	127	Auditors	14

The same is applied for target tokens:

Table 7 – Example of Target Forward Index

Segment ID	Location	Token	Number
0	2	الأمم (al-umam/nations)	0
0	8	المتحدة (al-muttaheda/united)	1
1	16	التقرير (al-taqrir/report)	2
1	23	المالي (al-maaly/financial)	3
1	30	والبيانات (w-al-bayanaat/ and statements)	4
1	38	المالية (al-maaleya/financial)	5
1	46	المراجعة (al-muraaja’ah/audited)	6
2	58	فترة (fatrah/period or biennium)	7
2	63	السنتين (al-sanatayn-biennium)	8
2	71	المنتهية (al-muntaheyah/ended)	9
2	78	31	10
2	81	كانون (kanum/December)	11
2	86	الأول (al-awwal/first)	12
2	93	ديسمبر (December /December)	13

2	98	2009	14
3	111	وتقرير (w-taqrir/and report)	15
3	116	مجلس (majles/council or board)	16
3	121	مراجعي (muraji'ee/revisers or auditors)	17
3	128	الحسابات (al-hesabaat/accounts)	18

We notice here that the word "financial" was mentioned more than once, with a unique index of each instance. We notice for the Arabic tokens that the word "report" corresponds to the tokens "التقرير (al-taqrir/report)" and "وتقرير (w-taqrir/and report)", which requires normalizing Arabic words to the base token.

3- Token normalization

In this step we convert all English words to lower case and stem the Arabic tokens to their base form. A simple Arabic morphological analyzer/stemmer was developed for this task (available online on arbsq.net/dev/my.cgi).

4- Inverted Indexing

The forwarded indexes are sorted and grouped by token, in the following way for example, and the target tokens are grouped the same way:

Table 8 – Example of an Inverted Index

Token	Segment ID	Token Location	Token Number in Forward Index
Papers	314	56468	5258
	314	56484	5261
	1485	342886	31920
Paris	710	182405	16584
	892	239027	21577
	898	240626	21731
Parts	434	89377	8399
	710	182207	16563
	1002	276936	24983

5- Creating Correspondence Dictionary

From the grouped list of tokens, and our word pair lexicon, we create correspondences between source tokens and target tokens. For language pairs which are similar, this

correspondence can be established simply by using cognates, as was done by Melamed; however for our case of English-Arabic pair, we cannot use cognates, so the following matching criteria is used:

i- If source token is the same as target token (in numbers and symbols and untranslated tokens), the two tokens are matching.
ii- If the source token and target token constitute an entry in the word pair lexicon, they are matching
iii- (experimental) we tried to use a transliteration scheme to match proper nouns, but it was not efficient enough in terms of processing time, but it can be useful in documents full of such nouns
iv- (experimental) multi-token word pairs, such as "New York/نيويورك" and "Auditors/مراجعي الحسابات", can be combined into one token, following the initial tokenization step; however, this task involves also more processing, since every Arabic token would need to be morphologically analyzed very early on, rather than after grouping all such tokens together.

So, we create a correspondence dictionary (in python) for all the source tokens. The keys of this dictionary are the source tokens, and the values are the corresponding target indexes for the matching tokens. The following is the algorithm for creating the dictionary:

```
Initialize correspondence_dictionary
for token1 in source_tokens:
        for token2 in target_tokens:
                if token1 and token2 are matching:
                        get token2 indexes
                        correspondence_dictionary[token1]= token2 indexes
```

6- Getting correspondence points for each token

For example, for the word "Paris", its grouped (inverted) indexes and the corresponding target token is "باريس" are the following:

Table 9 – Matching Inverted Indexes

Token	Segment ID	Token Location	Token Number in Forward Index
Paris	710	182405	16584
	892	239027	21577
	898	240626	21731
باریس	723	151065	20734
	905	197555	27151
	911	199246	27368

So, we have the following possible correspondence points:

Table 10 – Matching Inverted Indexes Coordinates

x-coordinate (source token location from the beginning)	y-coordinate (corresponding target token location from the beginning)
182405	151065
182405	197555
182405	199246
239027	151065
239027	197555
239027	199246
240626	151065
240626	197555
240626	199246

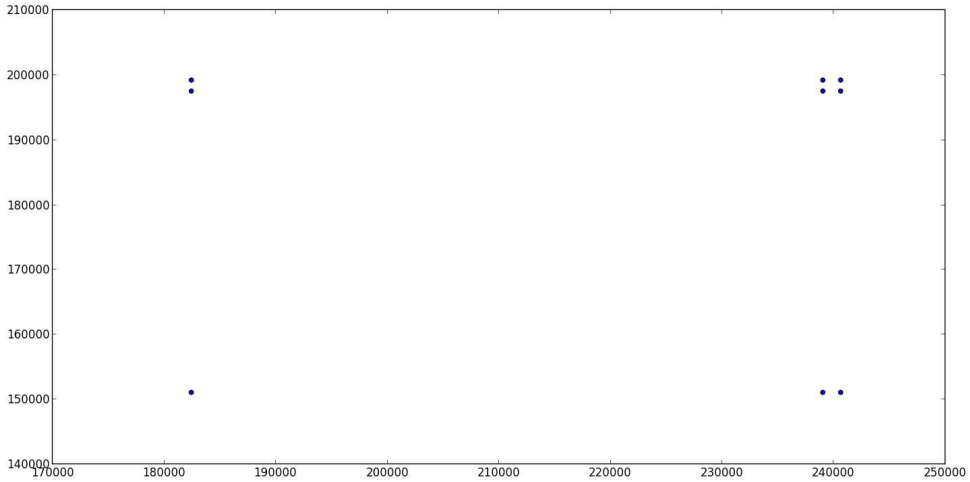

Figure 11 - Possible correspondence points of the word “Paris” and its equivalents

At this point, we proceed to investigate the probability of each of these points being a True Correspondence Point (TCP), meaning that this particular source token at this particular location corresponds to this particular target token at its particular location.

7- Evaluating likelihood of correspondence points

The approach used in evaluating the point likelihood to be a TCP is based on a hypothesis that the correct correspondence points are the ones which have a cluster of correspondence points nearby. So, we pick each possible point, whose coordinates are x0,y0, and identify the neighboring tokens, whose x-coordinate (x1) falls with a certain distance from the x-coordinate of our point (x0).

distance=75
min_x= x0 - distance
max_x= x0 + distance
neighboring_tokens=[token for x1[token]>min_x and x1[token]<max_x

Then we investigate the corresponding y-coordinates for each of these neighboring tokens, using the correspondence dictionary defined in step 5 above:

corresponding_y_coordinates= correspondence_dictionary[token] for token in neighboring_tokens

Now we need to measure the clustering around our point, so we identify for each token, the closest corresponding y-coordinates to our y-coordinate (y0).

y1=argmin abs(yi-y0) for i → 0, len(corresponding_y_coordinates[token[i]]

So, now we have the coordinates (x1,y1) for each neighboring token, then we identify a clustering factor as follows:

```
function get_point_distance([a1,b1],[a2,b2]):
        x_distance=abs(a2-a1)
        y_distance=abs(b2-b1)
        distance=sqrt(x_distance^2+x_distance^2)
        return distance

clstering_factor=Sigma (1/get_point_distance([x0,y0],[x1(token),y1(token)])
                For token in neighboring tokens
```

This way, the clustering factor is higher with more neighboring points close to our point, and it is also higher if the distance between these points and our point is smaller. We identify a threshold (0.1) for the clustering factor, and any point with a clustering factor above this threshold is accepted as candidate point for later processing.

8- Collect candidate points

We start this task by sorting the grouped source tokens by the token frequency. Then we filter out the tokens with the highest frequency (we filtered out 1/10 of the tokens), and proceed with the remaining (lower frequency) tokens one by one. The high frequency tokens are excluded because they require more processing time and can create much noise. We identify the candidate points for each token and add them to the list of all candidate points.

9- Identify point transitions for candidate points

Having collected the points with the highest clustering factor (candidate points), we identify possible transitions between points, as in the following example:

We have the following candidate points:

[1,5],[1,9],[1,48],[1,102],[4,9],[4,19],[4,50],[8,12],[8,22],[12,55],[12,140],[12,201], [19,29],[19,40],[23,170],[23,230]

We need to identify what are the transition possibilities for each of these points, so we do sorting and grouping by the x-coordinate, to have the following groups:

Group[0]=[1,5],[1,9],[1,48],[1,102]
Group[1]=[4,9],[4,19],[4,50]
Group[2]=[8,12],[8,22]
Group[3]=[12,55],[12,140],[12,201]
Group[4]=[19,29],[19,40]
Group[5]=[23,170],[23,230]

So, the possible transitions for any point are to any of points within the following three groups, as shown in the algorithm below:

```
Initialize point_transitions
for i in range(0,number of groups):
        current_points=Group[i]
        transition_points=Group[i+1]+Group[i+2]+Group[i+3]
        for point in current_points:
                point_tranisitions[point]= transition_points
```

So, in our example, the point [4,9] will have the corresponding transition points:

[8,12],[8,22], [12,55],[12,140],[12,201], [19,29],[19,40]

From these point transitions, we proceed to calculate the shortest path through these points.

10- Identify the shortest path within the candidate points

We use Dijkstra's algorithm to identify the shortest path within the candidate points, which would identify the TCPs that can be eventually used to identify alignment. To apply this algorithm, we need to have the start point (the origin), the end point (the termium), and the point transition dictionary which indicates the distance between each two points, and the algorithm would output the sequence of points which have the least cumulative shortest distance. However, if we use the geometrical distance, the algorithm would skip many points because it would favor having fewer points with shorter linear distance, and would be susceptible to following wrong paths if there are two consecutive false candidate points. For this reason, we use a modified distance for the transition between any two points, based on the distance function developed in step 7:

Modified distance=-1/get_point_distance(point1,point2)

The combined use of negative and inverse distance is to force Dijkstra's algorithm to follow the path which has more points having the least distance between them.
This is because the more points we have, the more negative the cumulative distance would be, and hence it will be the minimum distance the algorithm seeks. The inverse of the distance would punish the point transitions with large geometric distance.

Eventually, Dijkstra's algorithm yields the list of TCP's that would be the backbone for aligning the bitext.

11- Interpolate over gaps

Despite the modifications to the distance in Dijkstra's algorithm to favor choosing more points, there would be still many left-over points that need to be aligned. For these points, we identify gaps within the Dijkstra's output, and identify the points enclosed by these gaps, and then further identify their point transitions and apply Dijkstra's algorithm on them once again to get an interpolated list of points, to be added to the final list of TCPs.

12- Point pruning and simple point filling heuristic

After we do all the possible interpolations, we identify the segment pairs which correspond to the points identified as TCP's. Some of the points are just wrong, because they involve sharp spikes, or because there is unreasonable ratio between segment lengths. These points are identified and removed. For the segments for which no corresponding segments have been identified, we can resort to a simple heuristic to fill them out, as in the following example:

The following segments have been identified:
[0=0]
[1=1]
[4=4]
[5=6,7]
[6=8]
[8=11]
[9=12]

[14=14]

For any two consecutive points [x1,y1],[x2,y2]:

```
x_gap= x2-x1
y_gap= y2-y1
If x_gap=y_gap:
        For i in range(1,x_gap):
                Add point [x1+i,y1+i]
If x_gap=1 and y_gap<5:
        Add point [x1+1,range(y1+1,y1+y_gap)]
If y_gap=1 and x_gap<5:
        Add point [range(x1+1,x1+x_gap),y1+1]
```

This will allow us to add the following pairs of segment ID's:

[2=2]
[3=3]
[7=9,10]
[10,11,12,13=13]

So, we add these points to our final list of pairs of segment ID's, which are the final answer to the alignment problem.

Experimental Setup

In this paper, we are comparing the output of our alignment approach (Canvas) to two other alignment approaches.

1- Gale-Church Alignment:

This is a Python implementation for the Gale-Church alignment algorithm (Gale and Church, 1991), available from:
http://code.google.com/p/gachalign/downloads/detail?name=GACHALIGN.tar.gz&can=2&q=.

This algorithm determines sentence alignment based on length distribution independently from the language pair. However, the documents handled by this algorithm must be divided into sections that can be mapped together. Since this is not the case for the United Nations documents being studied, we manually create artificial sections to hypothetically test the performance of this alignment approach. We also experiment with the document as one large section to see if this approach can handle this (more real-world) case.

2- Champollion Alignment Toolkit:

This is a Perl implementation of the algorithm developed by (Ma, 2006), which combines the use of lexical elements with some penalty for the length.
This package is available from:
http://champollion.sourceforge.net/

In order to compare the performance of the three systems (Canvas, Gale-Church, Champollion), it is necessary to have evaluation bitexts with annotations for the segment correspondences. Although it is possible to find manually aligned documents of any size, we are interested mainly in having unaligned documents, which are annotated to indicate their correct alignment. This annotation is not an easy task, especially for large documents (greater than 1500 segments). So we annotated two documents below this size threshold, to indicate the various misalignment patterns exhibited (e.g. one-to-many, positive and negative offset, etc.).

For large documents, our main concern was to compare the speed and accuracy of the systems without due regard to these misalignment patterns, so we worked with already manually aligned documents, knowing simply that any deviation from their segment correspondences would indicate some inaccuracy. It should be noted, however, that using aligned documents mean that the bitext path follows the diagonal completely, and hence gives more advantage to other alignment approaches which checks segment similarity around the diagonal by specifying certain window size.

Results and Discussion

The table below indicates the results of the different alignment approaches utilized, when applied to four different documents with two main features. The first feature is whether the sections of the document pairs are displaced, causing complex misalignment patters such as positive and negative offset and spiked misalignment introduced above. The second feature is whether the document is already aligned, to facilitate measuring alignment performance for larger documents.

Table 11 – Experimental Results

Document ID		1	2	3	4
Number of Segments	Source	3282	218	1456	3648
	Target	3282	282	1434	3648
Document Features	Sections Displaced	No	Yes	No	No
	Already Manually Aligned	Yes	No	No	Yes
Processing Time	Gale-Church (No Sections)	23 min	5.2 sec	NA	NA
	Gale-Church (Sections)	18 min	0.53 sec	NA	NA
	Champollion	49 min	~ 1 min	8 min	29 min
	Canvas	176.3 sec	5.1 sec	21.6 sec	217.4 sec
Accuracy	Gale-Church (No Sections)	0%	0%	NA	NA
	Gale-Church (Sections)	~ 100% *	~ 88%*	NA	NA
	Champollion	97.9%	30.4%	76.3%	96.9%
	Canvas	96.6%	78.5%	97.1%	94.6%
Coverage	Gale-Church (No Sections)	0%	0%	NA	NA
	Gale-Church (Sections)	~ 100% *	~ 50%*	NA	NA
	Champollion	98.9%	90.6%	87.9%	96.0%
	Canvas	94.7%	79.6%	75.1%	88.2%
Accuracy* Coverage	Gale-Church (No Sections)	0%	0%	NA	NA
	Gale-Church (Sections)	~ 100% *	44.0%	NA	NA
	Champollion	96.8%	27.5%	67.1%	93.0%
	Canvas	91.4%	62.6%	72.9%	83.4%

~ Approximate
* Indicates that the result is based on manual inspection of the aligned sections.

The performance criteria consisted mainly of the following:

1- Processing Time: The time spent to align the document
2- Accuracy: The number of correct segment pairs obtained by the alignment system divided by the total number of segment pairs obtained
3- Coverage: The number of obtained segment pairs divided by the number of all correct segment pairs

4- Accuracy * Coverage: An empirical indicator to measure the alignment performance both in terms of accuracy and coverage

In terms of processing time, we found that our alignment system (Canvas), is faster by orders of magnitude than the other lexical-based system (Champollion). The speed ratio varied between 8 times to even more than 16 times. Comparing Canvas to a purely length-based approach such as Gale and Church, we had to handle two situations. The first is when the document is not split into sections, and the second is when it is split into sections. Without sections, Gale and Church processing time is similar to that of Canvas for smaller documents (document #2 in our test documents), where both were around 5 seconds. As for larger documents which are not split into sections, the processing time for Gale and Church is much higher than Canvas (about six times for document #1).

As for the accuracy, it appears that Canvas produced better accuracy than Champollion for documents which have intricate misalignment patterns, such as displaced sections (for document #2 which exhibits such patterns the accuracy of Canvas is almost twice that of Champollion). In larger documents which are already manually aligned, the accuracy of Canvas and Champollion are very similar, but the accuracy of Champollion is better, probably because it uses a window of segments around the diagonal so it would filter out distant segments.

Comparing the accuracy of Canvas to Gale and Church; however, is like comparing apples to oranges. For Gale and Church, the case is always that if the sections are well aligned, the accuracy is very good, though it would fail to detect the spiked misalignment due to alphabetical ordering of the countries, so its accuracy in such situations is around 88%, which is still higher than that of Canvas 78.5 %. However, it should always be considered that some carful manual alignment work was done to create the sections. If the document is treated as one section, both the accuracy and coverage of Gale and Church are simply zero.

We devised a custom indicator for how good the alignment system performs, by multiplying the accuracy with the coverage. This indicator was better for Canvas in the

case of unaligned documents; however, it was better for Champollion for aligned documents.

Conclusion

Sentence Alignment is not an easy task, as it entails many challenges, both general and specific to the language pair in question (English-Arabic), and to the domain (United Nations documents). Our alignment system (Canvas) was able to handle many of these misalignment patterns, and it performed better in terms of speed and accuracy than other lexical-based alignment systems (Champollion). The main drawback of the Canvas System is its coverage, where it is typically lower than Champollion and hence it will miss many segments. However, the improvement in speed can give many opportunities for improving both accuracy and coverage, by including more checks or better interpolation schemes, which can be a subject for future research. One of the main contributions of this project, in addition to the alignment system, is the creation of a nucleus of annotated unaligned material that can help investigate the rich diversity of misalignment patterns and identify better ways to handle them.

Bibliography

Brown, Peter F., Jennifer C. Lai, and Robert 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, California, USA, June. Association for Computational Linguistics.

Butt , Miriam, Dyvik , Helge, Holloway King, Tracy, Masuichi , Hiroshi, and Rohrer, Christian. 2002. The parallel grammar project. In Proceedings of COLING2002, Workshop on Grammar Engineering and Evaluation, pages 1–7.

Tsai , Chen-hui, Choi , HoJung. 2005. Parallel Corpora and Chinese Lexical Learning. In Proceedings of 4th International Conference on Internet Chinese Education. http://edu.ocac.gov.tw/icice2005/icice2005/html/paper2/C13.pdf

Chen, Stanley F. 1993. Aligning sentences in bilingual corpora using lexical information. In Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, pages 9–16, Columbus, Ohio, USA, June. Association for Computational Linguistics.

DeNero ,John and Uszkoreit , Jakob. 2011. Inducing Sentence Structure from Parallel Corpora for Reordering, In Proceedings of the Conference on Empirical Methods in Natural Language Processing.

Gale, W.A., Church, K.W.: A program for Aligning Sentences in Bilingual Corpora. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, California (1991) 177–184

Gomes, L. (2009) “Parallel Texts Alignment”, Master Degree thesis, Universidade Nova de Lisboa, Lisboa, Portugal.

Kay, Martin, and Röscheisen, Martin. "Text-translation alignment."*computational Linguistics* 19.1 (1993): 121-142.

Khadivi, S. 2008. Statistical Computer-Assisted Translation. Ph.D. thesis, RWTH-Aachen University, Aachen, Germany, July.

Li, Peng, Sun, Maosong, and Xue, Ping. 2010. Fast Champollion: A fast and robust sentence alignment algorithm. In 23rd International Conference on Computational Linguistics: Posters, pages 710–718, Beijing. Association for Computational Linguistics.

Ma, Xiaoyi. 2006. Champollion: A robust parallel text sentence aligner. In Proceedings of LREC- 2006: Fifth International Conference on Language Resources and Evaluation, pages 489–492.

Melamed, I. Dan. 1996. A geometric approach to mapping bitext correspondence. In Proceedings of the First Conference on Empirical Methods in Natural Language Processing, pages 1–12.

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

Simard, Michel, Foster, George F., and Isabelle, Pierre. 1993. Using cognates to align sentences in bilingual corpora. In Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative Research, pages 1071–1082. IBM Press.

Singh, Anil Kumar and Husain, Samar. (2005). Comparison, selection and use of sentence alignment algorithms for new language pairs. In Proceedings of the ACL Workshop on Building and using Parallel texts, pages 99–106.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., Varga, D.: The JRC-Acquis: A multilingual aligned parallel corpus with 20+languages. In: LREC 2006 (2006)

Tomeh, Nadi. Discriminative Alignment Models For Statistical Machine Translation. Diss. Université Paris Sud-Paris XI, 2012.

Wu, D.: Aligning a Parallel English-Chinese Corpus Statistically with Lexical Criteria. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico (1994) 80–87

Xiao, Richard. "What can SLA learn from contrastive corpus linguistics? The case of passive constructions in Chinese learner English." Indonesian Journal of English Language Teaching 3.2 (2007): 1-19.