

Automatic Construction of
Chinese-English Translation Lexicons

Final Project Report
under NSA grant
MDA904-97-C-3055

I. Dan Melamed and Mitch Marcus
Dept. of Computer and Information Science
University of Pennsylvania
Philadelphia, PA, 19104, U.S.A.
{melamed,marcus}@linc.cis.upenn.edu

Introduction and Executive Summary

The process of constructing translation lexicons from parallel texts (bitexts) can be broken down into three stages: mapping bitext correspondence, counting co-occurrences, and estimating a translation model. State-of-the-art techniques for accomplishing each stage of the process had already been developed, but only for bitexts involving fairly similar languages. Correct and efficient implementation of each stage poses special challenges when the parallel texts involve two very different languages. This report describes our theoretical and empirical investigations into how existing techniques might be extended and applied to Chinese/English bitexts.

Section 1 of the report describes our experience in porting the SIMR bitext mapping algorithm to Chinese/English. Contrary to popular skepticism, the syntactic and orthographic differences between Chinese and English did not pose a problem for SIMR. Objective evaluation showed that, after appropriate re-optimization, SIMR performed as well on Chinese/English bitexts as it did on bitexts in much more similar language pairs, such as French/English. Readers who are not familiar with SIMR or bitext geometry are urged to first read Appendix A and/or Melamed (1997), before reading Section 1 of this report.

Counting co-occurrences is more difficult for Chinese/English bitexts than for French/English bitexts, because the Chinese notion of a sentence is very different from the European notion. Chinese sentences are typically translated as several English sentences, and many-to-many correspondences are quite common (Xu & Tan, 1996). Therefore, counting co-occurrences in aligned segments is very inefficient for Chinese/English bitexts. Section 2 of the report shows how a more general co-occurrence model can be based on a more general bitext map. The new co-occurrence model does not presume knowledge of segment boundaries, and is therefore more suitable for bitexts involving languages as different as Chinese and English. Section 2 also exposes and corrects a common mistake in published methods for counting co-occurrences.

An encapsulated model of co-occurrence has the additional benefit of making word-to-word translation models independent of word order variations between the two halves of a bitext. This property, together with recent progress in automatic segmentation of Chinese text (Guo, 1998), implies that the third stage of the lexicon construction process can be applied exactly the same way to Chinese/English bitexts as to French/English bitexts. However, it is well known by lexicographers that word-to-word translations are just the tip of the iceberg, and that the more difficult challenge is to capture translational equivalents for “non-compositional compound” (NCC) words like *hot dog* and *kick the bucket*. The problem of finding minimal content-bearing units in text is most acute for languages that are written without spaces, such as Chinese, but it exists in all languages. A major part of our work was to develop fully automatic methods for identifying NCCs in bitext and for estimating their translational distributions. The method and results were documented in the project’s mid-term report, which is attached as Appendix B.

1 Chinese/English Bitext Maps

The report on porting SIMR to new language pairs (Melamed, 1996) recommends a three-step porting process: choose a matching predicate, implement axis generators, re-optimize SIMR's parameters. The way we carried out these three steps for Chinese/English is described in Sections 1.2, 1.3, and 1.4, respectively. First, a few words about the data we used.

1.1 Data

The initial plan was to use parallel Chinese/English news briefs from the China News Service. As the project got under way, we discovered that the news briefs were not suitable for our purposes because they were not really parallel texts. Rather, the Chinese and English articles in each pair were independently written by different authors, albeit on the same topic. Both articles in each pair were originals; neither was a translation of the other. Therefore, some of the assumptions made by our bitext mapping techniques would be invalid with respect to these data. For example, there was no guarantee that the order in which ideas were presented in an English text was anything like the order in the corresponding Chinese text.

Fortunately, we discovered that Resnik *et al.* (1997) had compiled the most common sixty-six books of the Bible online in a variety of languages, including Chinese and modern English. The Bible bitexts were more suitable for our project for two reasons, in addition to being truly parallel texts:

- The 66 books of the Bible exhibit a great variety of styles and genres. Working with multiple types of text has the advantage of more robust, more reliable, and more informative results.
- One of the steps in porting SIMR to a new language pair is the creation of training data for optimizing SIMR's parameters. Normally, this step involves the manual creation of bitext maps for a number of training bitexts. However, the Bible is already segmented into verses, and the segmentation is constant across all languages¹. Verse boundaries can serve as a ready-made, indisputable and fairly detailed bitext map. Thus, the Bible can be readily used for parameter optimization without laborious and time-consuming manual effort.

1.2 The Matching Predicate

The Chinese and English languages do not share an alphabet, so orthographic lexical cognates do not exist between them. A pre-existing translation lexicon must be used to get a sufficiently strong signal in a Chinese/English bitext space. We happened to have such a lexicon, consisting of 7997 entries. The Chinese/English matching predicate deemed two tokens to match if they constituted an entry in the translation lexicon. The axis generator for Chinese (described below) helped the matching predicate to also compare Chinese/English punctuation and numerals. In addition to consulting the translation lexicon, the matching predicate returned TRUE on any two tokens that were identical.

1.3 The Axis Generator for Chinese

Since orthographic lexical cognates do not exist between Chinese and English, the Chinese axis generator need not worry about identifying Chinese words. The only way a Chinese word might

¹Resnik *et al.* (1997) discuss exceptions.

be hypothesized to be a translation of an English word is if the two appeared on the opposite sides of an entry in our Chinese-English translation lexicon. Thus, the Chinese axis generator generated most of each Chinese axis by simple string matching: Every sequence of Chinese characters that appeared as the Chinese side of some lexicon entry was assigned its own axis position.

The remainder of each axis consisted of punctuation marks and numerals. These types of text tokens are often identical across languages, even when the languages have different writing systems in every other respect. In attempting to match punctuation marks between Chinese and English, we discovered a previously unforeseen problem: Even though the punctuation marks had the same appearance in both languages when printed, their electronic versions were encoded using different character sets! The English text was in plain ASCII. The Chinese texts, on the other hand, were in GB code, and even common punctuation marks such as commas and periods had their own two-byte GB encodings. Fortunately, the set of punctuation marks is quite small. To resolve the character set incompatibility, we wrote a pre-processor for Chinese text that converts the two-byte punctuation marks to their ASCII codes.

Numerals presented another problem. Although some Chinese texts use the Arabic system to express numerals, we discovered that there exists a Chinese number system too. Our Chinese Bible expressed its numerals using the Chinese system. In Chinese numerals, the position of a digit does not completely determine its tens exponent, like it does in the Arabic system, so the conversion algorithm could not simply substitute digits one-for-one. Fortunately, Chinese numerals are deterministically identifiable in Chinese text. To enable matching of numerals between Chinese and English, we added another module to our Chinese pre-processor. This module identifies Chinese numerals in the text, computes their numeric value, and re-expresses this value in Arabic digits. The Chinese (GB) pre-processing script is publicly available from <http://www.cis.upenn.edu/~melamed/>.

1.4 Parameter Optimization

The report on porting SIMR to new language pairs recommends re-optimizing SIMR’s parameters on bitext maps of at least 500 points. To obtain a more robust parameter set, we decided to run three separate optimizations, with the aim of cross-validation. For this purpose, we randomly selected three books of the Bible that had between 500 and 1000 verses: Proverbs, Mark and II Samuel. SIMR’s parameters were independently optimized on these three bitexts using simulated annealing. Each of the three resulting parameter sets was then tested on the other two bitexts. SIMR’s performance on the training data is shown in Table 1. The results that were obtained on

bitext	# of verses	Parameter Set 1	Parameter Set 2	Parameter Set 3
Mark	678	<i>7.89</i>	8.78	9.37
Proverbs	915	9.42	<i>6.46</i>	7.53
II Samuel	695	14.10	9.94	<i>7.81</i>

Table 1: *SIMR’s RMS error on Chinese/English training bitexts.*

the three initial optimizations are italicized. SIMR’s performance seemed to degrade the least with parameter set 2, so this parameter set was retained for testing.

1.5 Evaluation

We wanted to test SIMR on a variety of bitexts as quickly as possible. Therefore, for testing, we decided to use all the books of the Bible that consisted of less than 500 verses, of which there were ten. The evaluation metric, as usual, was the root mean squared distance, in characters, between TPCs (pairs of verse boundaries known to correspond) and the interpolated bitext map produced by SIMR, where the distance was measured perpendicular to the main diagonal. The results are given in Table 2.

bitext	# of verses	RMS Error
I Corinthians	437	17
II Corinthians	257	11
Daniel	357	12
Ecclesiastes	222	9.1
Ezra	280	59
Hebrews	303	14
Nehemiah	406	50
Revelations	404	6.8
Romans	433	14
Zechariah	211	15

Table 2: *SIMR's test results on Chinese/English bitexts.*

These results may be misleading in the following way. The TBM samples used for training and testing were derived from verse alignments. The alignments were converted into sets of co-ordinates in the bitext space by pairing the character positions at the ends of aligned segment pairs. Most of the aligned segments consisted of whole sentences, which end with a period. Wherever SIMR matched the periods correctly, the interpolated bitext map was pulled close to the TPC, even though it may have been much farther off in the middle of the sentence. Thus, this TBM sampling method artificially reduced the error estimates. The results in Table 2 should be considered only relative to each other and to other results obtained under the same experimental conditions.

With the exception of Ezra and Nehemiah, SIMR's accuracy on all the books was comparable to its accuracy on bitexts in other language pairs and other genres (Melamed, 1997). The two exceptions were surprising and disappointing. Bitext mapping is typically used in a pipeline with other processes, rather than as an end in itself, so it is unacceptable for a bitext mapping algorithm to fail two times out of ten. We undertook some error analysis in order to learn more.

The porting guidelines (Melamed, 1996) describe a general error-hunting strategy that can be used to debug training data intended for optimizing SIMR's parameters. We used this strategy to analyze our test data. We found that SIMR got both the Ezra and Nehemiah bitext maps mostly right. However, both of the problematic bitexts contained short segments where SIMR made several large errors in a row. The RMS error metric is very sensitive to large errors, even if they are few in number.

Next, we looked at the bitexts themselves, in the region where SIMR was far off the mark. The problem became readily apparent. The Bible contains a number of passages describing genealogies. For instance, Figure 1 contains an excerpt from the New International Version of the book of Ezra. The genealogy text segments contain little besides names and punctuation. The names are not in our translation lexicon. The punctuation marks alone do not provide a sufficiently strong signal for SIMR to follow. SIMR essentially skipped over these regions of the bitext space, unable to find

10:38: From the descendants of Binnui: Shimei,
10:39: Shelemiah, Nathan, Adaiah,
10:40: Macnadebai, Shashai, Sharai,
10:41: Azarel, Shelemiah, Shemariah,
10:42: Shallum, Amariah and Joseph.
10:43: From the descendants of Nebo: Jeiel, Mattithiah, Zabad, Zebina, Jaddai, Joel
and Benaiah.

Figure 1: *Genealogy excerpt from the book of Ezra.*

any suitable chains of correspondence points there. Interpolation of SIMR’s bitext map across the gaps was a poor approximation to the TBM for these bitexts.

The reason for SIMR’s poor performance on Ezra and on Nehemiah gives us two reasons to be optimistic. First, the problem arose from a quirk in the bitext – exceedingly long strings of proper nouns – that is rare outside the Bible. Second, the problem is solvable. Proper nouns that are translated into Chinese from another language are usually written so as to retain much of their pronunciation. Therefore, proper nouns can be matched across Chinese/English bitexts as phonetic cognates (Melamed, 1997). A number of recent empirical studies have shown the feasibility of this approach (Knight & Graehl, 1997; Chen *et al.*, 1998; Wan & Verspoor, 1998). A matching predicate that can supplement its translation lexicon with phonetic cognates could provide an even stronger signal. In addition to preventing large errors in the rare case encountered here, the stronger signal can improve SIMR’s performance on more typical bitexts.

2 Models of Co-occurrence

A **model of co-occurrence** is a boolean predicate, which indicates whether a given pair of word *tokens* co-occur in corresponding regions of the bitext space. Co-occurrence is a precondition for the possibility that two tokens might be mutual translations. Models of co-occurrence are the glue that binds methods for mapping bitext correspondence with methods for estimating translation models into an integrated system for exploiting parallel texts. When the model of co-occurrence is modularized away from the translation model, it also becomes easier to study translation model estimation methods *per se*.

Most methods for estimating translation models from bitext start with the following intuition: Words that are translations of each other are more likely to appear in corresponding bitext regions than other pairs of words. The intuition is simple, but its correct exploitation turns out to be rather subtle. Most of the literature on translation model estimation presumes that corresponding regions of the input bitexts are represented by neatly aligned segments. However, aligning Chinese/English bitexts is of dubious utility, because the two languages have very different notions of how to group text into segments, and so their corresponding segments are typically not very informative to translation model estimation algorithms. Moreover, imposing an alignment relation on bitexts is inefficient, because alignments cannot capture crossing correspondences between text segments.

Different models of co-occurrence are possible, depending on the kind of bitext map that is available, the language-specific information that is available, and the assumptions made about the nature of translational equivalence. The following three sections explore these three variables.

2.1 Relevant Regions of the Bitext Space

Corresponding regions of a text and its translation will contain word token pairs that are mutual translations, by definition of “mutual translations.” Therefore, a general representation of bitext correspondence is the natural framework in which to ground a model of where mutual translations co-occur. The most general representation of bitext correspondence is a bitext map. Token pairs whose co-ordinates are part of the true bitext map (TBM) are mutual translations, by definition of the TBM. The likelihood that two tokens are mutual translations is inversely correlated with the distance between the tokens’ co-ordinate in the bitext space and the interpolated TBM.

It may be possible to develop translation model estimation methods that take into account a probabilistic model of co-occurrence. However, all the models in the literature so far are based on a boolean co-occurrence model — they want to know either that two tokens co-occur or that they do not. A boolean co-occurrence predicate can be defined in several ways. Most researchers interested in co-occurrence of mutual translations have relied on bitexts where sentence boundaries (or other text unit boundaries) were easy to find (*e.g.* Gale & Church, 1991b; Kumano & Hirakawa, 1994; Fung, 1995; Melamed, 1995). Aligned text segments suggest a **boundary-based model of co-occurrence**, illustrated in Figure 2. For the reasons given above, this model is unsuitable for Chinese/English bitexts.

A more general approach, which does not rely on knowledge of segment boundaries, is to set a threshold δ on the distance from the interpolated bitext map. Any token pair whose co-ordinate is closer than δ to the bitext map would be considered to co-occur by this predicate. The optimal value of δ varies with the language pair, the bitext genre and the application. Figure 3 illustrates what I will call the **distance-based model of co-occurrence**. Dagan *et al.* (1993) were the first to use a distance-based model of co-occurrence, although they measured the distance in words rather than in characters.

For bitexts involving languages with similar word order, a more accurate **combined model of co-occurrence** can be built using both segment boundary information and the map-distance threshold. As shown in Figure 4, each of these constraints eliminates the noise from a characteristic region of the bitext space.

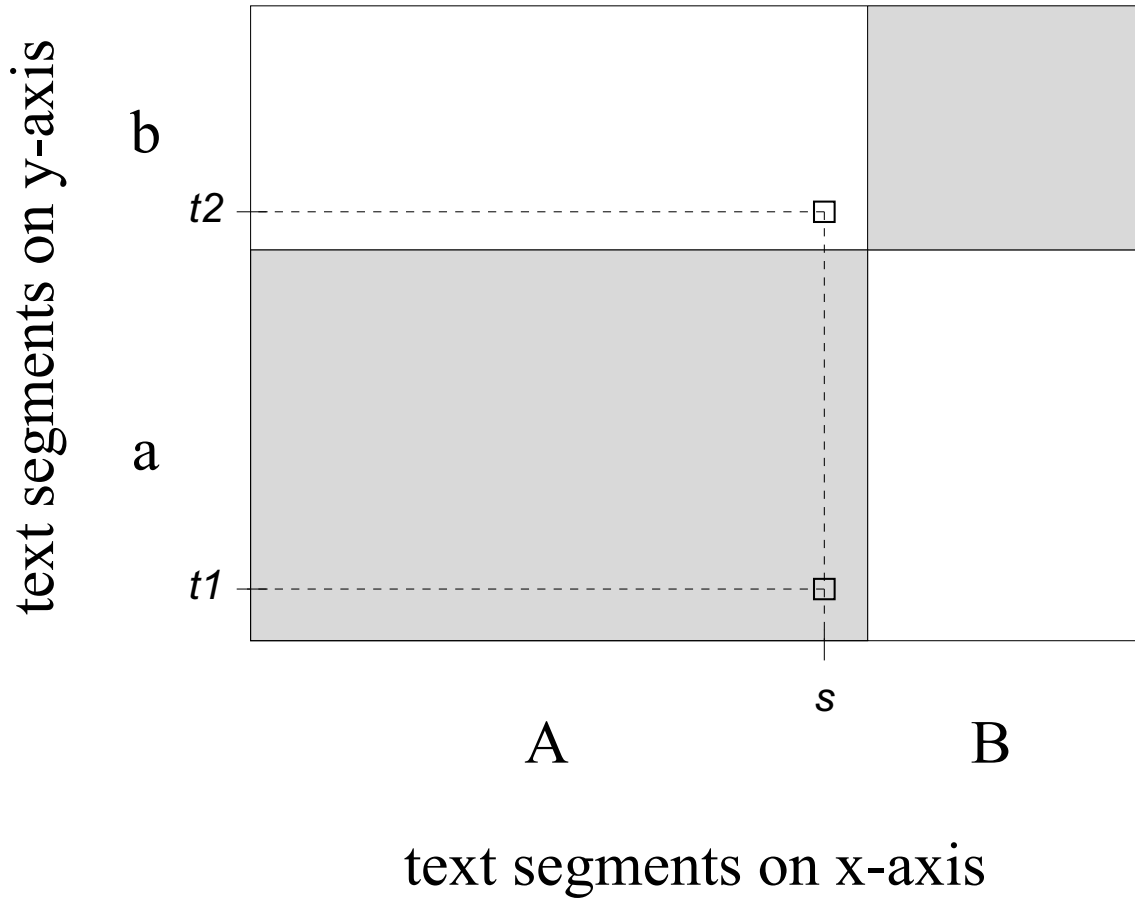


Figure 2: *Boundary-based model of co-occurrence. Word token pairs whose co-ordinates lie in shaded regions count as co-occurrences. Thus, $(s, t1)$ co-occur, but $(s, t2)$ do not.*

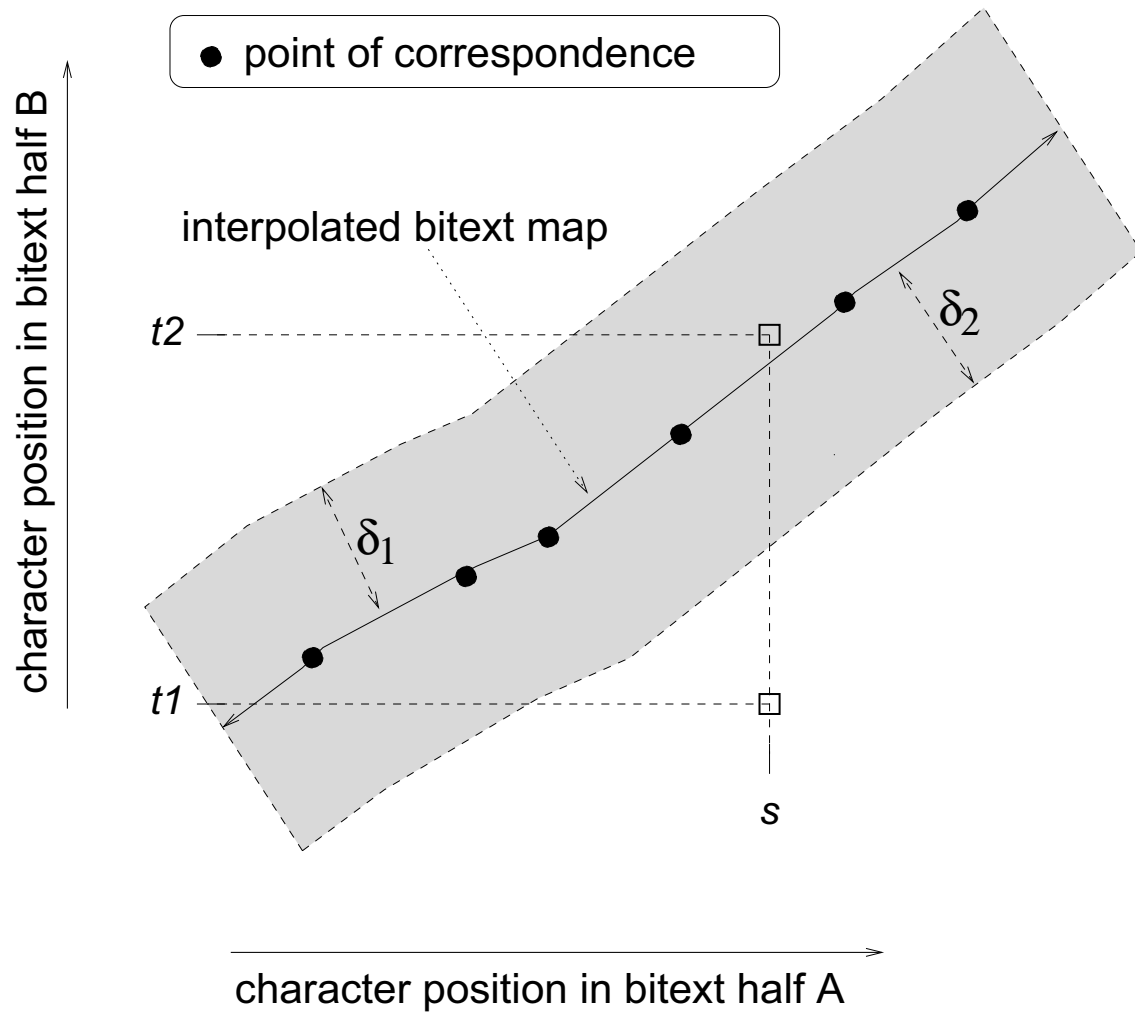


Figure 3: *Distance-based model of co-occurrence.* Word token pairs whose co-ordinates lie in the shaded region count as co-occurrences. In contrast with Figure 2, (s, t_2) co-occur, but (s, t_1) do not.

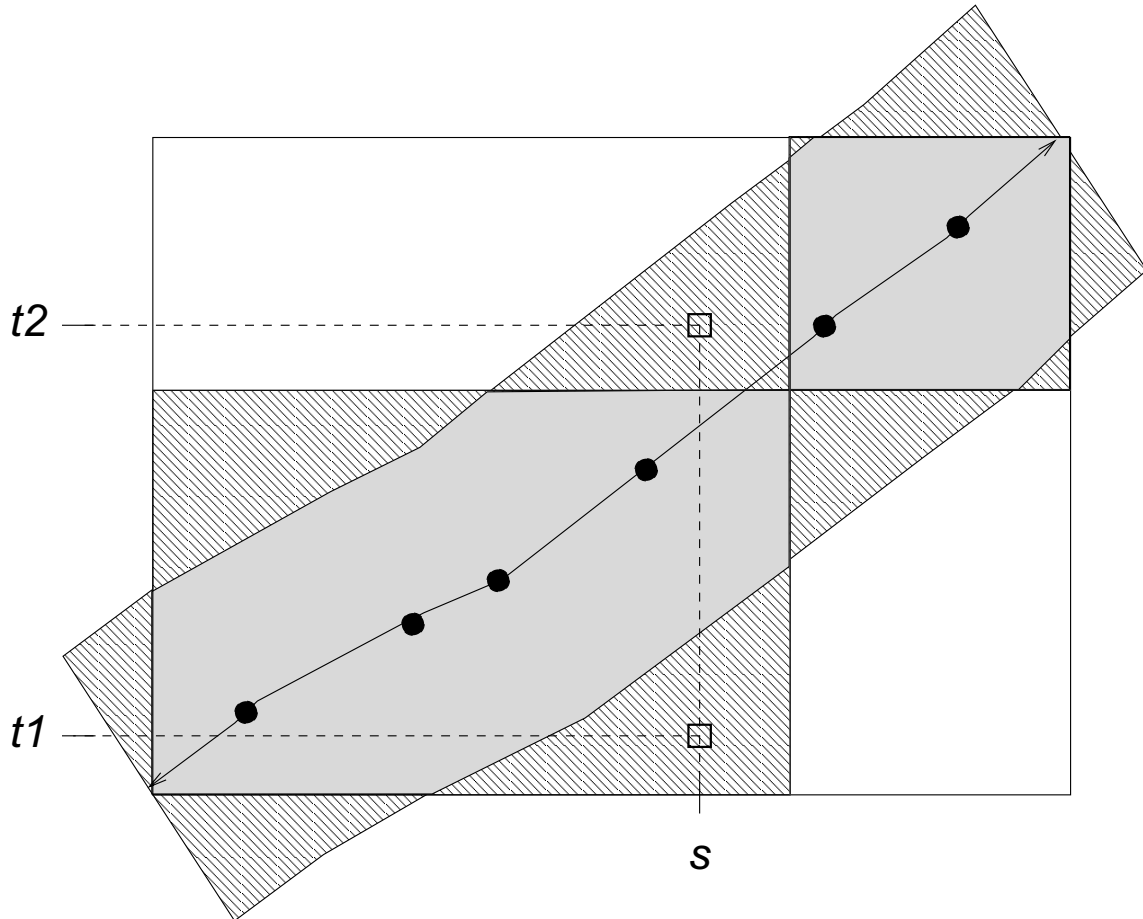


Figure 4: *Combined model of co-occurrence. Word token pairs whose co-ordinates lie in shaded regions count as co-occurrences. In contrast with Figures 2 and 3, neither $(s, t1)$ nor $(s, t2)$ co-occur. Striped regions indicate eliminated sources of noise.*

2.2 Co-occurrence Counting Methods

Both the boundary-based and distance-based constraints restrict the region of the bitext space where tokens may be considered to co-occur. These constraints do not tell us how to count co-occurrences within the restricted regions. It is somewhat surprising that this is a question at all, and most researchers ignore it. However, when authors specify their algorithms in sufficient detail to answer this question, the most common answer (given, *e.g.*, by Brown *et al.*, 1993; Dagan *et al.*, 1993; Kupiec, 1993; Melamed, 1995) turns out to be unsound. The problem is easiest to illustrate under the boundary-based model of co-occurrence. Given two aligned text segments, the naive way to count co-occurrences is

$$cooc(\mathbf{u}, \mathbf{v}) = e(\mathbf{u}) \cdot f(\mathbf{v}) \quad (1)$$

where $e(\mathbf{u})$ and $f(\mathbf{v})$ are the frequencies of occurrence of \mathbf{u} and \mathbf{v} in their respective segments. For many \mathbf{u} and \mathbf{v} , $e(\mathbf{u})$ and $f(\mathbf{v})$ are either 0 or 1, and Equation 1 returns 1 just in case both words occur. The problem arises when $e(\mathbf{u}) > 1$ and $f(\mathbf{v}) > 1$. For example, if $e(\mathbf{u}) = f(\mathbf{v}) = 3$, then according to Equation 1, $cooc(\mathbf{u}, \mathbf{v}) = 9!$ If the two aligned segments are really translations of each other, then it is most likely that each of the occurrences of \mathbf{u} is a translation of just one of the occurrences of \mathbf{v} . Although it may not be known which of the 3 v 's each u corresponds to, the number of times that \mathbf{u} and \mathbf{v} co-occur as possible translations of each other in that segment pair must be 3.

There are various ways to arrive at $cooc(\mathbf{u}, \mathbf{v}) = 3$. Two of the simplest ways are

$$cooc(\mathbf{u}, \mathbf{v}) = \min[e(\mathbf{u}), f(\mathbf{v})] \quad (2)$$

and

$$cooc(\mathbf{u}, \mathbf{v}) = \max[e(\mathbf{u}), f(\mathbf{v})]. \quad (3)$$

Equation 2 is based on the simplifying assumption that each word is translated to at most one other word. Equation 3 is based on the simplifying assumption that each word is translated to at least one other word. Either simplifying assumption results in more plausible co-occurrence counts than the naive method in Equation 1.

Counting co-occurrences is more difficult under a distance-based co-occurrence model, because there are no aligned segments and consequently no useful definition for $e()$ and $f()$. Furthermore, under a distance-based co-occurrence model, the co-occurrence relation is not transitive. *E.g.*, it is possible that s_1 co-occurs with t_1 , t_1 co-occurs with s_2 , s_2 co-occurs with t_2 , but s_1 does not co-occur with t_2 . The correct counting method becomes clearer if the problem is recast in graph-theoretic terms. Let the words in each half of the bitext represent the vertices on one side of a bipartite graph. Let there be edges between each pair of words whose co-ordinates are closer than δ to the bitext map. Now, under the “at most one” assumption of Equation 2, each co-occurrence is represented by an edge in the graph’s maximum matching². Under the “at least one” assumption of Equation 3, each co-occurrence is represented by an edge in the graph’s smallest vertex cover. Maximum matching can be computed in polynomial time for any graph (Ahuja *et al.*, 1993). Vertex cover can be solved in polynomial time for bipartite graphs³. It is of no importance that maximum matchings and minimum vertex covers may be non-unique — by definition, all solutions have the same number of edges, and this number is the correct co-occurrence count.

²A **maximum matching** is a subgraph that solves the cardinality matching problem (Ahuja *et al.*, 1993, pp. 469-470).

³The algorithm is folklore, but Phillips & Warnow (1996) describe relevant methods.

2.3 Language-Specific Filters

Co-occurrence is a universal precondition for translational equivalence among word tokens in bitexts. Other preconditions may be imposed if certain language-specific resources are available (Melamed, 1995). For example, parts of speech tend to be preserved in translation (Papageorgiou *et al.*, 1994). If part-of-speech taggers are available for both languages in a bitext, and if cases where one part of speech is translated to another are not important for the intended application, then we can rule out the possibility of translational equivalence for all token pairs involving different parts of speech. A more obvious source of language-specific information is a machine-readable bilingual dictionary (MRBD). If token a in one half of the bitext is found to co-occur with token b in the other half, and (a, b) is an entry in the MRBD, then it is highly likely that the tokens a and b are indeed mutual translations. In this case, there is no point considering the co-occurrence of a or b with any other token. Similarly, exclusive candidacy can be granted to cognate token pairs.

Conclusion

The project described in this report accomplished three tasks to advance the state of the art of automatic construction of translation lexicons. First, we showed that the SIMR bitext mapping algorithm is truly portable to any language pair. We draw this conclusion from the premise that if it works for English and Chinese, then it will certainly work for language pairs that are more similar. Second, we introduced models of co-occurrence, a theoretical framework for gluing together algorithms for mapping bitext correspondence with algorithms for estimating statistical translation models. Our theoretical innovation enabled us to discover and correct a flaw in the most commonly used co-occurrence counting methods. Third, we developed a principled information-theoretic method to identify non-compositional compound (NCC) words in bitext, and to model their translational distributions. Objective empirical evaluation clearly showed the advantages of such a principled approach.

As with most successful research projects, this one raised as many questions as it answered. We believe the bitext mapping problem to be largely solved, although it will take some work to optimize our research-speed software to run at production speed. It would also be useful to test the method on a wider variety of text genres. The models of co-occurrence that we have introduced are certainly not exhaustive. Other models of co-occurrence may be more efficient in their representation of the co-occurrence relation. Our models of translational equivalence made significant progress over prior work. However, they have yet to account for most of the vast complexity of language.

References

- R. K. Ahuja, T. L. Magnati & J. B. Orlin. (1993) *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ.
- P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, & R. L. Mercer. (1993) "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics* 19(2).
- H.-H. Chen, S.-J. Huang, Y.-W. Ding, & S.-C. Tsai. (1998) "Proper Name Translation in Cross-Language Information Retrieval," *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*. Montreal, Canada.

- I. Dagan, K. Church, & W. Gale. (1993) "Robust Word Alignment for Machine Aided Translation," *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*. Columbus, OH.
- P. Fung. (1995) "A Pattern Matching Method for Finding Noun and Proper Noun Translations from Noisy Parallel Corpora," *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. Boston, MA.
- W. Gale, & K. W. Church. (1991a) "A Program for Aligning Sentences in Bilingual Corpora" *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*. Berkeley, CA.
- W. Gale & K. W. Church. (1991b) "Identifying Word Correspondences in Parallel Texts," *Proceedings of the DARPA SNL Workshop*. Asilomar, CA.
- J. Guo. (1998) "One Tokenization per Source," *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*. Montreal, Canada.
- K. Knight & J. Graehl. (1997) "Machine Transliteration," *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*. Madrid, Spain.
- A. Kumano & H. Hiraoka. (1994) "Building an MT Dictionary from Parallel Texts Based on Linguistic and Statistical Information," *Proceedings of the 15th International Conference on Computational Linguistics*. Kyoto, Japan.
- J. Kupiec. (1993) "An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora," *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*. Columbus, OH.
- I. D. Melamed. (1995) "Automatic evaluation and uniform filter cascades for inducing N -best translation lexicons,"
- I. D. Melamed. (1996) "Porting SIMR to new language pairs," IRCS Technical Report 96-26. University of Pennsylvania.
- I. D. Melamed. (1997) "A Portable Algorithm for Mapping Bitext Correspondence," *Proceedings of the 35th Conference of the Association for Computational Linguistics*. Madrid, Spain.
- H. Papageorgiou, L. Cranias & S. Piperidis. (1994) "Automatic Alignment in Parallel Corpora," *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (Student Session)*. Las Cruces, NM.
- C. Phillips & T. J. Warnow. (1996) "The Asymmetric median tree — A New Model for Building Consensus Trees," *Discrete Applied Mathematics* 71(1-3), pp. 331-335.
- P. Resnik, M. B. Olsen & M. Diab. (1997) "Creating a Parallel Corpus from the Book of 2000 Tongues," *Proceedings of the 10th TEI User Conference*. Providence, RI.
- S. Wan & M. Verspoor. (1998) "Automatic English-Chinese Name Transliteration for Development of Multilingual Resources," *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*. Montreal, Canada.
- D. Xu & C. L. Tan. (1996) "Automatic Alignment of English-Chinese Bilingual Texts of CNS News," *Proceedings of the 1996 International Conference on Chinese Computing*.

A Bitext Geometry

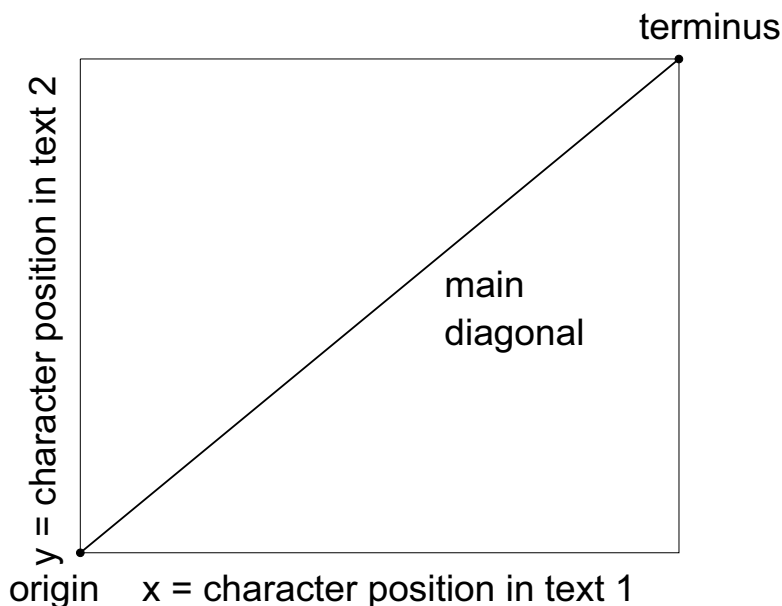


Figure 5: A *bitext space*.

Each bitext defines a rectangular **bitext space**, as illustrated in Figure 5. The lower left corner of the rectangle is the **origin** of the bitext space and represents the two texts' beginnings. The upper right corner is the **terminus** and represents the texts' ends. The line between the origin and the terminus is the **main diagonal**. The slope of the main diagonal is the **bitext slope**.

Each bitext space is spanned by a pair of **axes**. The lengths of the axes are the lengths of the two component texts. The axes of a bitext space are measured in characters, because text lengths measured in characters correlate better than text lengths measured in tokens (Gale & Church, 1991a). This correlation is important for geometric bitext mapping heuristics. Although the axes are measured in characters, I will argue that word tokens are the optimum level of analysis for bitext mapping. By convention, each token is assigned the position of its median character.

Each bitext space contains a number of **true points of correspondence (TPCs)**, other than the origin and the terminus. TPCs exist both at the co-ordinates of matching text units and at the co-ordinates of matching text unit boundaries. If a token at position p on the x-axis and a token at position q on the y-axis are translations of each other, then the coordinate (p, q) in the bitext space is a TPC. If a sentence on the x-axis ends at character r and the corresponding sentence on the y-axis ends at character s , then the co-ordinate $(r + .5, s + .5)$ is a TPC. The $.5$ is added because it is the inter-sentence boundaries that correspond, rather than the last characters of the sentences. Similarly, TPCs arise from corresponding boundaries between paragraphs, chapters, list items, *etc.*. Groups of TPCs with a roughly linear arrangement in the bitext space are called **chains**.

Bitext maps are injective (1-to-1) partial functions in bitext spaces. A complete set of TPCs for a particular bitext is the **true bitext map (TBM)**. The purpose of a **bitext mapping algorithm** is to produce bitext maps that are the best possible approximations of each bitext's TBM.

B The Mid-term Project Report