**Automatic Alignment for Scripture**

*Andi Wu*

*Global Bible Initiative*

**Abstract**

A word-to-word alignment of the scripture to its translations is required for many applications in Bible engagement and Bible translation. It must be done automatically to make the applications feasible. This paper presents a hybrid approach of automatic alignment where a pure statistical word aligner is used in conjunction with a linguistic knowledge base of the original Hebrew and Greek texts. Knowledge bases of this kind are not generally available, but they exist in the Bible domain and can be leveraged to significantly improve the performance of the aligner. Evaluations show that the use of morphological and syntactic knowledge does take the accuracy to a new level. The aligner is language-independent and can be used to align any word-segmented scripture texts.

1. **Introduction**

Alignment, the process whereby texts are mapped word-to-word between source and target language texts, has been proven valuable in Bible translation. It is a prerequisite for many applications, from key-term checking to the building of interlinears, concordances, translation memories, and statistical models of computer-assisted translation. As manual alignment is costly and time-consuming, automatic alignment by the machine is required to meet the needs of these applications.

In recent years, significant progress has been made in statistical methods of automatic alignment. These methods work well on large text corpora and can produce results that are good enough for certain applications such as statistical machine translation. However, Scripture alignment has less data to work with and requires higher accuracy. In the experiments to be presented in this paper, we augment statistical methods with linguistic knowledge of the source texts: lemmatization alleviates the sparse data problem, parts of speech help to identify content words, and syntactic trees assist in resolving potential ambiguities.

In what follows, we will first discuss the evaluation metrics and use them to evaluate the results of a pure statistical aligner. Then we will experiment with the use of morphological and syntactic knowledge in the alignment process and see the improvement in accuracy.

1. **Evaluation Metrics**

In order to evaluate the performance of an aligner and track the progress we are making, we need for each text a “gold standard” – a “perfect” alignment – against which the machine-aligned output can be compared. Fortunately, Global Bible Initiative (GBI) is able to provide such gold standards in multiple versions that cover 7 different languages: English, Chinese, Japanese, Hindi, Punjabi, Burmese, and Khmer. Over the years, GBI has created a big database of aligned texts whose accuracy has been manually checked word by word. For the purpose of this experiment, the following versions are selected in the following languages. (The versions marked with \* are translated by GBI.)

* English
  + ESV (English Standard Version)
  + NASB95 (New American Standard Bible 1995)
  + KJV (King James Version)
  + LEB (Lexham English Bible)
  + NRSV (New Revised Standard Version)
  + NIV84 (New International Version 1984)
* Chinese
  + CUV (Chinese Union Version)
  + RCUV (Revised Chinese Union Version)
  + NCV (New Chinese Version)
  + CSB (Chinese Standard Bible) \*
* Japanese
  + SKN (Masao Sekine Translation)
  + SHK (New Japanese Bible)
* Hindi
  + HOVR (Hindi Old Version Re-edited)
  + HSB (Hindi Standard Bible) \*
* Punjabi
  + PSB (Punjabi Standard Bible) \*
* Burmese
  + JVB (Judson Version Burmese Bible)
  + MCL (Myanmar Common Language Bible)
  + MSB (Myanmar Standard Bible) \*
* Khmer
  + KHOV (Khmer Old Version 1954)
  + KHSV (Khmer Standard Version)

We run the automatic aligner on all these versions and then compare the links in the output with the links in the gold standards. A link is a correspondence between a source word and a target word, where the target word is supposed to be the translation of the source word. A perfect alignment should contain all the links found the gold standard and not contain any link not found in the gold standard. Therefore we use the following three metrics to measure the accuracy of an alignment:

1. Recall. Of all the links in the gold standard, how many of them are found in the automatic alignment? The score is the percentage of the links in the gold standard which are also found in the alignment:
2. Precision. Of all the links in the automatic alignment, how many of them are found in the gold standard? The score is the percentage of the links in the alignment which are also found the gold standard:
3. F-score. This is a harmonic mean of Recall and Precision. This balances recall and precision which often work against each other:

For example, if the links in the gold standard are A, C, D, F, G while the links in the automatic alignment are A, B, C, E, G, H, recall will be 60% and precision will be 50%, with an F-score of 55%.

1. **Using a pure statistical aligner**

We started with a pure statistical aligner implemented by Damien Daspit of SIL. The aligner trains direct and inverse HMM-based single word alignment models[[1]](#footnote-1). It takes two parallel files and produces an alignment. For each segment pair, which is a verse in our case, it computes the best alignment using each model and merges them together. It may also take an optional auxiliary file which is an existing partial alignment that can guide the current alignment.

In our first experiment, we ran the aligner without the auxiliary file, as no partial alignment was available. The results are used as a baseline for aligner performance. The aligner was run on all the versions listed in the last section, but due to limited space in this paper, we are not able to show the scores of all versions. Instead, we will pick one version from each language and show its scores[[2]](#footnote-2):

|  |  |  |
| --- | --- | --- |
| English | Recall | 72.43% |
| (ESV) | Precision | 57.77% |
|  | F-Score | **64.27%** |
| Chinese | Recall | 61.82% |
| (CSB) | Precision | 48.35% |
|  | F-Score | **54.24%** |
| Japanese | Recall | 51.93% |
| (SHK) | Precision | 41.48% |
|  | F-Score | **46.08%** |
| Hindi | Recall | 58.14% |
| (HSB) | Precision | 43.29% |
|  | F-Score | **49.62%** |
| Punjabi | Recall | 54.46% |
| (PSB) | Precision | 39.86% |
|  | F-Score | **46.03%** |
| Burmese | Recall | 51.01% |
| (MSB) | Precision | 41.00% |
|  | F-Score | **45.46%** |
| Khmer | Recall | 58.52% |
| (KHOV) | Precision | 43.43% |
|  | F-Score | **49.84%** |

To improve upon these results, we run the aligner twice. An auxiliary file is created from the first alignment by taking those links whose probabilities are above a given threshold. The threshold is empirically determined through experiments. The auxiliary file is then used to guide the second alignment. This boosted the scores significantly, as we can see in the following comparison.

|  |  |  |  |
| --- | --- | --- | --- |
| **Language** | **Metric** | **Without Aux File** | **With Aux File** |
| English | Recall | 73.52% | 80.25% |
| (ESV) | Precision | 57.95% | 69.11% |
|  | F-Score | **64.81%** | **74.27%** |
| Chinese | Recall | 61.82% | 68.45% |
| (CSB) | Precision | 48.35% | 55.51% |
|  | F-Score | **54.24%** | **61.28%** |
| Japanese | Recall | 51.93% | 57.61% |
| (SHK) | Precision | 41.48% | 45.97% |
|  | F-Score | **46.08%** | **51.11%** |
| Hindi | Recall | 58.14% | 65.16% |
| (HSB) | Precision | 43.29% | 52.11% |
|  | F-Score | **49.62%** | **57.91%** |
| Punjabi | Recall | 54.46% | 61.96% |
| (PSB) | Precision | 39.86% | 49.23% |
|  | F-Score | **46.03%** | **54.86%** |
| Burmese | Recall | 51.01% | 57.94% |
| (MSB) | Precision | 41.00% | 44.81% |
|  | F-Score | **45.46%** | **50.54%** |
| Khmer | Recall | 58.52% | 66.81% |
| (KHOV) | Precision | 43.43% | 50.16% |
|  | F-Score | **49.84%** | **57.30%** |

As we can see, the use of an auxiliary file, which identifies the more reliable links first and then uses them to guide the alignment of the remaining words, has a very positive effect on the results. This two-pass approach is used in all the experiments to be discussed later.

However, these scores are still far from the accuracy we require for aligning the scripture, but this is the best a pure statistical aligner can do with the data we have. It performs much better in the general domain where the corpora to be aligned are much bigger, but such “big data” is not available in the Bible domain.

While we are at a disadvantage as far as data size is concerned, we do have data that is not commonly available in the general domain. We have rich data on the source side where the Hebrew and Greek texts have been analyzed morphologically and syntactically. The data is available from GBI’s linguistic knowledge base. We should be able to take advantage of the data and use it to improve automatic alignment. In other worlds, what we have in terms of data quality may compensate for our lack in data quantity. In what follows, we will show how the linguistic data can help raise the accuracy of alignment.

1. **Using root forms instead of surface forms**

An obvious solution to the problem of small data size is to lower the type-token ratio (TTR) of the text.  TTR is obtained by dividing the types (the total number of different/unique words) occurring in a text by its tokens (the total number of words). A lower TTR indicates a lower degree of lexical variation, which makes automatic alignment easier. We can lower the TTR by using the root forms of words (lemmas) as text tokens instead of surface forms. The TTR is lower because many different surface forms are now identified as being the same lemma.

The identification of lemmas requires morphological analysis which is not feasible in the general domain, but the analysis has already been done on the source side. The target side has no such analysis, of course, and the tokens have to be the surface forms. But even using lemmas only on the source side has a big impact, as we see below, where the only difference is that lemmas are used instead of surface forms on the Hebrew and Greek side.

|  |  |  |  |
| --- | --- | --- | --- |
| **Language** | **Metric** | **Surface forms on both sides** | **Lemmas on source side** |
| English | Recall | 80.25% | 88.35% |
| (ESV) | Precision | 69.11% | 76.56% |
|  | F-Score | **74.27%** | **82.03%** |
| Chinese | Recall | 68.45% | 81.85% |
| (CSB) | Precision | 55.51% | 65.13% |
|  | F-Score | **61.28%** | **72.54%** |
| Japanese | Recall | 57.61% | 72.90% |
| (SHK) | Precision | 45.97% | 56.07% |
|  | F-Score | **51.11%** | **63.38%** |
| Hindi | Recall | 65.16% | 74.66% |
| (HSB) | Precision | 52.11% | 63.10% |
|  | F-Score | **57.91%** | **68.40%** |
| Punjabi | Recall | 61.96% | 73.28% |
| (PSB) | Precision | 49.23% | 61.25% |
|  | F-Score | **54.86%** | **66.73%** |
| Burmese | Recall | 57.94% | 72.08% |
| (MSB) | Precision | 44.81% | 54.40% |
|  | F-Score | **50.54%** | **62.00%** |
| Khmer | Recall | 66.81% | 80.14% |
| (KHOV) | Precision | 50.16% | 62.46% |
|  | F-Score | **57.30%** | **70.19%** |

The change in the type-token ratio yields a big improvement. The increase in percentage is more than 10 points in most languages. This does not require any linguistic analysis of the target text at all. The linguistic analysis done on the source side can benefit the alignment regardless of what the target language is.

1. **Aligning content words only**

In spite of the improvement brought by the use of lemmas, the quality of alignment is still not satisfactory. One thing we notice is that precision is significantly lower than recall across the board. That means many of the links found by the machine are invalid. There is a lot of “junk” in the output. A close examination of the results shows that such junk is often associated with links that involve function words – articles, pronouns, prepositions, conjunctions, and sometimes adverbs – which tend to be noise in the data. The alignment should be much cleaner if such noise is filtered out. In other words, the accuracy should improve if we only align content words – nouns, verbs and adjectives.

For practical purposes, a good alignment of the content words should be sufficient for most applications in the Bible domain. The key terms to be checked, the words we need in a concordance, and the queries in a search usually consist of content words. If we can have high accuracy in aligning content words, most of the needs will be met. Therefore, we decided to focus on the alignment of content words only, at least for now.

In order to filter out the function words in the input, we need to know the part of speech (POS) of every word. Such information cannot be assumed in the general domain, but is available in the scripture. In the linguistically analyzed Hebrew and Greek texts we use, every word has a POS tag which tells us whether the word is a content word or function word. However, the target texts are not linguistically analyzed and do not have POS tags.

When we use lemmas in alignment, we are able to have lemmas on one side only and still get good results. Can we do the same and do alignment with content words only on the source side and all words on the target side? We did experiment with that, but it did not work. To align content words only, we must have content words only on both sides. This requires us to find a way to automatically identify content words on the target side. This led to a side project of automatic differentiation of content words and function words. We will discuss this side project in the next section. This project succeeded in identifying most of the function and content words in the target text. We then tried aligning the texts which content words only. As expected, the scores improved, as shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Language** | **Metric** | **All Words** | **Content Words only** |
| English | Recall | 88.35% | 89.07% |
| (ESV) | Precision | 76.56% | 88.41% |
|  | F-Score | **82.03%** | **88.73%** |
| Chinese | Recall | 81.85% | 88.08% |
| (CSB) | Precision | 65.13% | 84.77% |
|  | F-Score | **72.54%** | **86.39%** |
| Japanese | Recall | 72.90% | 81.06% |
| (SHK) | Precision | 56.07% | 78.39% |
|  | F-Score | **63.38%** | **79.68%** |
| Hindi | Recall | 74.66% | 83.51% |
| (HSB) | Precision | 63.10% | 76.47% |
|  | F-Score | **68.40%** | **79.83%** |
| Punjabi | Recall | 73.28% | 80.64% |
| (PSB) | Precision | 61.25% | 87.23% |
|  | F-Score | **66.73%** | **77.61%** |
| Burmese | Recall | 72.08% | 68.01% |
| (MSB) | Precision | 54.40% | 63.78% |
|  | F-Score | **62.00%** | **65.83%** |
| Khmer | Recall | 80.14% | 83.42% |
| (KHOV) | Precision | 62.46% | 70.70% |
|  | F-Score | **70.19%** | **76.54%** |

The improvement is quite significant, especially in languages other than English, with more gains in precision than recall. Using only content words further lowers the type-token ratio. Function words tend not to inflect and therefore using their lemmas does not change the ratio very much. Content words, on the other hand, can appear in many different inflected forms. Using lemmas instead of surface forms has a much bigger impact on content words.

1. **Identifying content words**

Content words and function words form two complementary sets. If a word is not a function word, then it is a content word. Therefore, if we can automatically identify the function words in a text, we will be able to identify the content words as well. So we started with identifying the function words in each language.

Function words have three characteristics. First of all, they form a closed set. There is only a limited number of function words in each language. Unlike content words, they can be exhaustively listed. Secondly, they are usually high-frequency words. Thirdly, their distribution is quite even: they are found everywhere, their occurrence hardly determined by the content of the text.

On the basis of these characteristics, we let the machine gather statistics from the texts of each language and produce a word list ordered by frequency and evenness of distribution. It turns out that most of the function words are found in the several hundred words at the top of the list.

However, some content words, such as “God”, “Lord”, and “Jesus”, are also high-frequency words and they are also evenly distributed. How do we prevent them from being mistakenly identified as function words? To solve this problem, we resorted to the use of an automatically generated gloss in each language. Using the log-likelihood-ratio algorithm for computing word association[[3]](#footnote-3), we can take two parallel files, just as we do in alignment, and produce an ordered list of bilingual word pairs, with the more reliable ones at the top.  The frequent content words usually appear in this list. As an example, here are the top 10 content word pairs found in English, Chinese and French:

English

2654 θεός God

2334 Ἰησοῦς Jesus

1713 Χριστός Christ

1467 κύριος Lord

1364 λέγω said

927 πίστις faith

903 μαθητής disciples

746 πατήρ Father

745 πνεῦμα Spirit

700 βασιλεία kingdom

Chinese

1366 λέγω 说

928 θεός 神

624 Ἰησοῦς 耶稣

595 κύριος 主

481 πατήρ 父

448 ἀποκρίνομαι 回答

419 Ἰωάννης 约翰

402 Φαρισαῖος 法利赛

394 μαθητής 门徒们

393 Πέτρος 彼得

French

3114 θεός Dieu

1858 κύριος Seigneur

1678 λέγω :

1624 Ἰησοῦς Jésus

892 μαθητής disciples

887 Χριστός Christ

836 πνεῦμα Esprit

812 πατήρ Père

777 ἀδελφός frères

736 γῆ terre

In those word pairs, the POS of the source side is already known, since all the Hebrew and Greek words are POS-tagged. Assuming that content words in Hebrew and Greek are translated into content words in the target language, we can identify a target word as a content word if its corresponding source word is a content word. Thus we can create a content word list from this gloss and use this to filter the list of potential function words. If a word occurs frequently, is distributed evenly, and is not in the list of frequent content words, it must be a function word.

Here are the top 10 function words found in English, Chinese and French:

English

the 0.54481582537517

and 0.406412005457026

to 0.346657571623465

of 0.298328785811733

you 0.179809004092769

in 0.177694406548431

that 0.13506139154161

is 0.133321964529332

not 0.126705320600273

I 0.12575034106412

Chinese

的 0.585684558418508

了 0.188628435032744

在 0.185109247811984

是 0.178040271742457

他 0.17620417406206

我 0.165983230307852

你们 0.158271620050187

就 0.156833343533876

不 0.138686578125956

他们 0.120754024114083

French

de 0.370755279009731

et 0.297610379294367

la 0.245382935063216

le 0.244621698550341

à 0.199311577414444

qui 0.188753557953267

il 0.17035149268551

les 0.167902296948435

est 0.164460184020653

vous 0.156351360296551

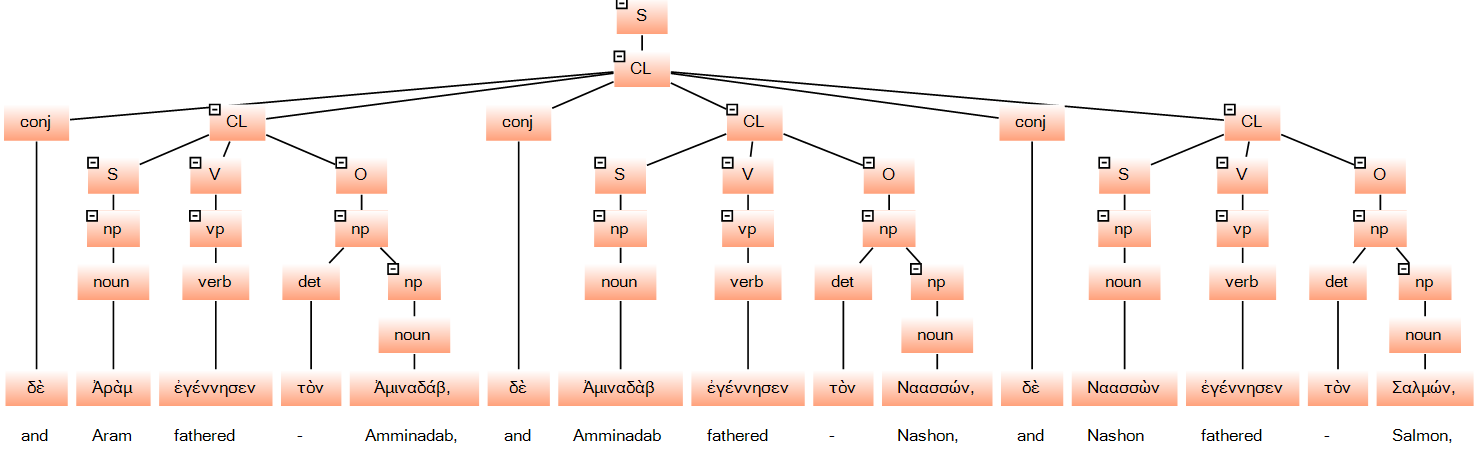
Now that we know what the function words are, we can filter out all the function words in the text and align the content words only. This is how we achieved the improvements shown in the previous section.

1. **Using syntactic trees in alignment**

As is true for most machine applications, it is not easy to improve the accuracy once it is in the 80% area. But we believed that the linguistic information we have can enable us to go a few extra miles. So we went a step further and tried making use of syntactic structures.

One observation from examining the output is that the aligner tends to go wrong when certain words occur multiple time in different positions in a verse, causing potential ambiguities in alignment. This problem is more prominent when the verse being aligned is long and the word order in the translation is very different. This is where syntactic trees can be of help.

Syntactic trees can be viewed as a multiple-level segmentation, where sentences are divided into clauses, clauses to phrases, and phrases to words. Each node in the tree represents a segment at a given level. In the following Greek tree, for example, the verse consists of three clauses as three segments of the verse.



In this verse, “Ἀμιναδάβ” and “Ναασσών” occur 2 times and “ἐγέννησεν” 3 times . Given the English translation “*and Ram the father of Amminadab, and Amminadab the father of Nahshon, and Nahshon the father of Salmon* “, the translation model of the aligner can tell us that Ἀμιναδάβ” should be linked to “Amminadab”, “Ναασσών “ to “Nahshon”, and ἐγέννησεν” to “father”. However, it does not tell us which “Ἀμιναδάβ” should be linked to which “Amminadab”, which “Ναασσών” to which “Nahshon”, or which “ἐγέννησεν” to which “father”. In the simple case here, where the translation has the same word order as the Greek original, the aligner can use position information to get the correct alignment. If the word order is scrambled in the translation (e.g. the positions of the second clause and the third clause are switched, though it is unlikely to happen in this case), the aligner can make mistakes.

This kind of mistakes can be avoided if we align clause by clause instead of the whole verse at once. We assume that each of the clauses correspond to a contiguous segment of text in the translation. In other words, the words in the same segment are close to each other. Then, for each segment in the source text, we want to align it to a target segment which not only contains the corresponding words, but has those words in close proximity.

Suppose we are trying to align the second clause in the above tree. For the sake of illustration, we will number some of the words in the translation:

*and Ram the father(1) of Amminadab(1), and Amminadab(2) the father(2) of Nahshon(1), and Nahshon(2) the father(3) of Salmon.*

According to the translation model, all the following links are possible. (The number in square brackets is the number of words between the first word in the alignment and the last word in the alignment.)

1. Amminadab(1) – father(1) – Nahshon(1) [6]
2. Amminadab(1) – father(1) – Nahshon(2) [8]
3. Amminadab(1) – father(2) – Nahshon(1) [4]
4. Amminadab(1) – father(2) – Nahshon(2) [6]
5. Amminadab(1) – father(3) – Nahshon(1) [7]
6. Amminadab(1) – father(3) – Nahshon(2) [7]
7. Amminadab(2) – father(1) – Nahshon(1) [6]
8. Amminadab(2) – father(1) – Nahshon(2) [8]
9. Amminadab(2) – father(2) – Nahshon(1) [2]
10. Amminadab(2) – father(2) – Nahshon(2) [4]
11. Amminadab(2) – father(3) – Nahshon(1) [5]
12. Amminadab(2) – father(3) – Nahshon(2) [4]

However, only the alignment in i) is the correct one and the distance between the first word and last word here is the shortest: 2. So the best alignment for a given phrase or clause in Hebrew and Greek is the target segment that not only has the highest joint translation probability but the shortest distance between its component words as well.

1. **Developing a tree-based aligner**

With the assurance that syntactic trees can help making alignment decisions, I developed a tree-based aligner which implements the idea illustrated in the previous section. It uses a translation model and a tree-based decoding system.

The translation model contains the translation probabilities of source-target word pairs. The probability of each word pair indicates the likelihood of the source word being translated into the target word. For the sake of simplicity, I use the original Dice coefficient index:

where *P* is the translation probability of a source-target word pair, the source word, the target word, the count of same-verse co-occurrence of *X* and *Y* in the aligned file, | the count of in the source file, and |*Y*| the count of *Y* in the target file.

During decoding, we traverse the tree bottom-up. For each terminal node (i.e. each single word), we get the translation probabilities of this word paired with each of the target word. The bilingual pairs are sorted by their probabilities and the top N are kept for further computation. For each non-terminal node, we get all the possible paths of combining the possible alignments of the child nodes, compute the joint probability and the distance (as defined in the previous section), adjust the joint probability with distance value, sort all the paths by their probabilities, and keep the top N paths. This goes on until we reach the root node, where the top path is picked as the alignment of the verse. In short, we do the alignment segment by segment and layer by layer, going from child nodes to parent nodes and picking the best N alignments at each layer, until the top layer is reached.

Before we run the tree-based aligner, we also run the pure statistical aligner and get the probability of each link in the output. This probability is used to adjust the probability from the translation model I built myself using the Dice coefficient index. The fact that a bilingual pair does end up as a link in the output of the pure statistical aligner means that the probability of the word pair being a true link is very high. So, if a word pair is found as a link in the output, we will increase the probability by an amount proportional to the probability of that link in the output of the statistical aligner. In this sense, the statistical aligner is used to guide the tree-based aligner. It helps to pick the best candidates at each level, reduce the search space, and speed up the tree-based aligner.

The results are interesting: there is a small loss in recall but there are big gains in precision, especially in Hindi, Punjabi, Burmese, and Khmer.

|  |  |  |  |
| --- | --- | --- | --- |
| **Language** | **Metric** | **Without Tree-Based Aligner** | **With Tree-Based Aligner** |
| English | Recall | 89.07% | 88.46% |
| (ESV) | Precision | 88.41% | 91.93% |
|  | F-Score | **88.73%** | **90.16%** |
| Chinese | Recall | 88.08% | 87.63% |
| (CSB) | Precision | 84.77% | 89.34% |
|  | F-Score | **86.39%** | **88.48%** |
| Japanese | Recall | 81.06% | 79.72% |
| (SHK) | Precision | 78.39% | 83.99% |
|  | F-Score | **79.68%** | **81.80%** |
| Hindi | Recall | 83.51% | 82.95% |
| (HSB) | Precision | 76.47% | 85.85% |
|  | F-Score | **79.83%** | **84.38%** |
| Punjabi | Recall | 80.64% | 80.04% |
| (PSB) | Precision | 74.79% | 83.44% |
|  | F-Score | **77.61%** | **81.70%** |
| Burmese | Recall | 68.01% | 67.16% |
| (MSB) | Precision | 63.78% | 76.37% |
|  | F-Score | **65.83%** | **71.47%** |
| Khmer | Recall | 83.42% | 81.13% |
| (KHOV) | Precision | 70.70% | 78.35% |
|  | F-Score | **76.54%** | **79.71%** |

The tree-based aligner gets fewer links but the quality of the links is much higher. There is less junk in the data.

The tree-based aligner is still being worked on, as there is still room for improvement. One advantage of the tree-based aligner is that it is not a black box. When things go wrong, we know exactly where the problem is. We not only see the changes in the scores, but also know why the scores changed. We can fine-tune the system without rebuilding the whole model.

1. **Aligning a minimal corpus**

Alignment is an important part of machine-assisted translation where the system learns from the aligned sentences. In Bible translation, especially in minority languages, the machine usually has very little data to learn from except the verses that have already been translated. The corpus to be aligned can be one chapter or less. We are therefore interested in knowing how well the aligners perform when they are used to align such minimal corpora.

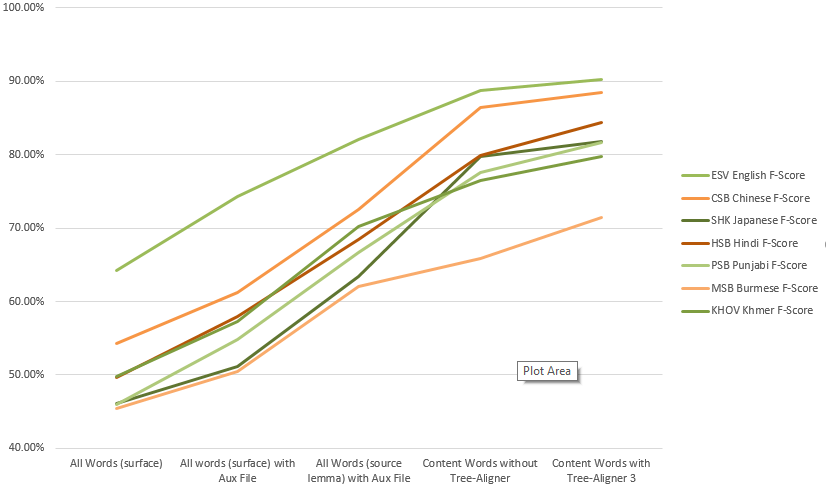
The accuracy of the aligners is expected to drop as the data size decreases, due to the statistical nature of the our methods. To find out how useful the aligners are in aligning minimal corpora, we picked the shortest book in the Bible – 2 John which only has 13 verses – and run the aligners on it in different languages. Here is the result:

|  |  |  |  |
| --- | --- | --- | --- |
| **Language** | **Metric** | **Without tree-based aligner** | **With tree-based aligner** |
| English | Recall | 46.74% | 51.09% |
| (ESV) | Precision | 44.44% | 55.81% |
|  | F-Score | **45.56%** | **53.35%** |
| Chinese | Recall | 51.61% | 63.44% |
| (CSB) | Precision | 45.71% | 64.84% |
|  | F-Score | **48.48%** | **64.13%** |
| Japanese | Recall | 46.67% | 38.89% |
| (SHK) | Precision | 34.43% | 41.67% |
|  | F-Score | **39.62%** | **40.23%** |
| Hindi | Recall | 52.13% | 53.19% |
| (HSB) | Precision | 39.37% | 57.47% |
|  | F-Score | **44.86%** | **55.25%** |
| Punjabi | Recall | 40.22% | 50.00% |
| (PSB) | Precision | 36.19% | 51.69% |
|  | F-Score | **38.10%** | **50.83%** |
| Burmese | Recall | 38.30% | 42.55% |
| (MSB) | Precision | 36.45% | 45.98% |
|  | F-Score | **37.35%** | **44.20%** |
| Khmer | Recall | 42.22% | 43.33% |
| (KHOV) | Precision | 30.65% | 43.33% |
|  | F-Score | **35.51%** | **43.33%** |

We see that it makes a big difference whether the tree-based aligner is used on top the pure statistical aligner. The tree-based aligner seems to depend less on data size than the pure statistical aligner. The use of linguistic knowledge seems to make it more robust when the data is statistically deficient.

1. **Conclusion**

Automatic alignment for scripture has the disadvantage of small data sizes but has the advantage of having the linguistic knowledge on the source side. By using the morphological and syntactic analysis already done in the Hebrew and Greek texts, we can significantly improve the performance of the statistical aligners. The following graph summarizes the progress we have made.



The more accurate output will advance the technology in both Bible engagement and Bible translation. Accessing the rich scripture data from any language or automatically checking the consistency of a Bible translation may become a reality.

1. See <http://www.aclweb.org/anthology/C/C96/C96-2141.pdf> for the models Damien implemented. [↑](#footnote-ref-1)
2. If the version has NT only, the score is that of the NT. Otherwise, it is ((OT\_Score \* 3) + NT\_Score) / 4, where OT is given more weight because its size is about 3 times of NT. [↑](#footnote-ref-2)
3. See “A Discriminative Framework for Bilingual Word Alignment” by Robert C. Moore at<http://dl.acm.org/citation.cfm?id=1220586> . [↑](#footnote-ref-3)