# Evaluating (and Improving) Sentence Alignment under Noisy Conditions

**Omar Zaidan**
Microsoft Research, USA
ozaidan@cs.jhu.edu

**Vishal Chowdhary**
Microsoft Research, USA
vishalc@microsoft.com

## Abstract

Sentence alignment is an important step in the preparation of parallel data. Most aligners do not perform very well when the input is a noisy, rather than a highly-parallel, document pair. Evaluating aligners under noisy conditions would seem to require creating an evaluation dataset by manually annotating a noisy document for gold-standard alignments. Such a costly process hinders our ability to evaluate an aligner under various types and levels of noise. In this paper, we propose a new evaluation framework for sentence aligners, which is particularly suitable for noisy-data evaluation. Our approach is unique as it requires no manual labeling, instead relying on small parallel datasets (already at the disposal of MT researchers) to generate many evaluation datasets that mimic a variety of noisy conditions. We use our framework to perform a comprehensive comparison of three aligners under noisy conditions. Furthermore, our framework facilitates the fine-tuning of a state-of-the-art sentence aligner, allowing us to substantially increase its recall rates by anywhere from 5% to 14% (absolute) across several language pairs.

## 1 Introduction

Virtually all training pipelines of statistical machine translation systems expect training data to be in the form of a sequence of parallel sentence pairs. This means that a pair of parallel documents must first be segmented into a sequence of aligned sentence pairs, discarding or combining sentences when needed, and aligning sentences as appropriate. The performance and output of an SMT system is directly dependent on the amount and quality of available training data. Therefore, it is critical to perform this *sentence alignment* step properly, ensuring both high recall (to have as much training data as possible) and high precision (to avoid noisy training data).

While sentence aligners achieve excellent performance on highly-parallel, clean data, the task is much more difficult under noisy conditions. Some prior work has investigated evaluation under noisy conditions (see section 6), but the major focus of prior work has been the clean-data scenario, where accuracy rates exceed 98% (e.g. Simard et al. (1993), Moore (2002)). For one thing, this meant that the various sentence alignment algorithms differ only slightly in absolute terms. Similarly, fine-tuning any one of those algorithms might not seem to have an impact on performance. More importantly, this also meant that we do not have a clear understanding of how well these algorithms would perform under noisy conditions.

Arguably, there was little need to examine sentence alignment of noisy datasets in early MT research, since almost all training data came from high-quality, highly-parallel sources, such as UN documents or parliamentary proceedings.[1] However, recent efforts have attempted to utilize web resources and non-perfectly-parallel texts, such as Wikipedia articles and news stories (e.g. Resnik and Smith (2003), Utiyama and Isahara (2003), Munteanu and Marcu (2005), and Smith et al. (2010)). Such resources naturally contain significantly more noise, at a level that would render sentence alignment a much less straightforward task.

Because sentence alignment algorithms had usually been evaluated under a clean-data scenario, there are fewer empirical results to guide those who wish to extract parallel data from noisy

[1] Also, parallel datasets created explicitly for MT research (by having a source corpus translated into the target language) would be already sentence-aligned by mere construction if the source side is split into sentences beforehand.



*Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 484–493,
Sofia, Bulgaria, August 8-9, 2013 

sources. Furthermore, there is also no easy way to fine-tune an aligner of interest. For building the Microsoft Translation service, we are continuously mining inherently-noisy web resources, from which we extract MT training data for dozens of the world's languages. Therefore, having a principled method to evaluate and fine-tune our aligner was critical.

In this paper, we describe our framework for evaluating sentence alignment under noisy conditions. We use this framework to examine and evaluate the Moore alignment algorithm (Moore, 2002), which was empirically shown to be state-of-the-art under clean conditions, and which we regularly use to extract parallel data from web resources to create training data. We perform a comprehensive comparison of this aligner against two other algorithms, and furthermore use our framework to fine-tune the algorithm along dimensions of interest (such as the aligner's search parameters) by quantitatively evaluating how the aligner's performance is affected by such changes.

The paper is organized as follows. We briefly define sentence alignment and existing approaches in section 2. We then discuss the *evaluation* of alignment algorithms in section 3, and present our evaluation framework. In section 4, we perform a comparative assessment of three alignment algorithms using our framework, illustrating the differences between them under noisy conditions. In section 5, we present two additional applications of our framework, namely fine-tuning an aligner and performing training data cleanup. Finally, we give an overview in section 6 of prior work that has tackled the specific issue of evaluating sentence aligners.

## 2 Sentence Alignment

*Sentence alignment* is the process by which a pair of parallel documents lacking explicit sentence links are used to extract a parallel dataset consisting of sentence pairs that are translations of each other. Specifically, let $S$ and $T$ be the document pair to be aligned, with $S$ composed of the sentence sequence $s_1, s_2, ..., s_m$, and $T$ composed of the sentence sequence $t_1, t_2, ..., t_n$. A sentence alignment of $S$ and $T$ is a segmentation of each of $S$ and $T$ into $p$ sequences $s'_1, s'_2, ..., s'_p$ and $t'_1, t'_2, ..., t'_p$ such that the following holds about the segmentation of $S$: (a similar set of conditions exist that correspond to $T$)

- $s'_i = C_S[a, b)$ for some $1 \leq a \leq b \leq m \; \forall i$
- $s'_1 = C_S[1, b)$ for some $b >= 1$
- $s'_p = C_S[a, m)$ for some $a <= m$
- If $s'_i = C_S[a, b)$, then $s'_{i+1} = C_S[b, c)$
- If $s'_i = C_S[x, x)$, then $t'_i = C_T[y, z)$ such that $y \neq z$

Above, $C_S[a, b)$ is the concatenation of $s_a, s_{a+1}, ..., s_{b-1}$, which indicates the possibility of aligning multiple source sentences to a single sentence (or combined sequence of sentences) on the target side. Note that $C_S[a, a)$ is the empty string, which indicates deletion on the target side (i.e. a target sentence is aligned to the empty string). The last condition disallows aligning an empty string to another empty string, thus eliminating the possibility for an infinite segmentation sequence.

Note that the result of this segmentation is $q$ (non-empty) sentence pairs, where $q <= p$ (and naturally $q <= m$ and $q <= n$). The deleted sentences, each aligned with an empty string, are left out of the resulting parallel corpus.

### 2.1 Approaches to Sentence Alignment

Tiedemann (2007) and Santos (2011) each provide a broad overview of sentence alignment, giving a timeline of relevant research and discussing algorithms and performance metrics for sentence alignment. In general, there are two main approaches to sentence alignment: length-based and lexical-based.

In length-based alignment approaches (e.g. Brown et al. (1991), Gale and Church (1991), and Kay and Röscheisen (1993)), the aligner relies on a probabilistic model that describes the source-to-target sentence length ratio for a pair of corresponding sentences. Such a model would account both for the average or typical length ratio as well as its variance. The aligner proceeds to align sentence pairs such that the output would be highly likely under the length ratio model.

In lexical-based alignment approaches (e.g. Chen (1993), Melamed (1997), Simard and Plamondon (1998), Menezes and Richardson (2001), and the LDC alignment tool, *Champollion* (Ma, 2006)), the aligner relies on a probabilistic model that describes the lexical similarity between a pair of sentences. The model could either be a fully-trained translation model, or a simpler bilingual

lexicon that finds corresponding word pairs. In contrast to length-based algorithms, lexical-based approaches typically require external bilingual resources, and usually perform better.

Previous work on sentence alignment varies across a few other dimensions as well. Some lexical-based algorithms build the needed bilingual resources from the very dataset that is to be aligned, whereas other approaches assume that such resources are externally provided. Another dimension is the need to provide anchor points within the text to be aligned, such as in the form of paragraph-level alignment. Such anchor points are typically needed to restrict the search space to a manageable size.

Another group of aligners take a hybrid approach, relying both on sentence length and lexical similarity (e.g. Zhao and Vogel (2002)). One notable example is the algorithm by Moore (2002), which has the benefit of relying only on the input data when training the lexical similarity model, rather than needing external resources (bilingual lexicon or parallel training data) for that purpose. The Moore algorithm is a state-of-the-art algorithm, and has been used, for example, to align the data for the Europarl corpus (Koehn, 2005), and is often a strong baseline in papers proposing new alignment algorithms (e.g. Braune and Fraser (2010)). In section 4, we use our proposed framework to evaluate Moore's algorithm, and compare it against two other aligners, illustrating our framework's utility as a comparative tool.

## 3 Evaluating Sentence Alignment Algorithms under Noisy Conditions

In much of the prior work mentioned above in 2.1, and in other comparative evaluation work (e.g. Simard et al. (1993), Langlais et al. (1998), and Véronis and Langlais (2000)), sentence alignment algorithms were evaluated using a manually-created gold-standard dataset. This is done by taking a parallel dataset, and manually annotating sentence pairs that are translations of each other (and should therefore be aligned). This evaluation dataset is provided as input to the aligner, which is evaluated based on the precision and recall of its output, as measured against the set of hand-annotated sentence pairs.

While this is a reasonable approach that mirrors the evaluation model in many other tasks within machine learning (i.e. to manually create an evaluation set with gold-standard labels, based on which the learner's output is judged), it suffers from some drawbacks.

For one thing, all the difficulties of creating an evaluation dataset apply here as well. Most significantly, manually labeling sentence pairs is costly and time-consuming. This problem is magnified in the context of machine translation, since one should ideally evaluate a sentence alignment algorithm under several language pairs, rather than a single one, requiring the creation of several evaluation sets, rather than a single one.

Furthermore, prior work usually used a fairly clean dataset to annotate, on which it is relatively easy for an aligner to achieve very high precision and recall rates. This means that differences between algorithms are sometimes fairly small in absolute terms, making it difficult to attribute such differences to the algorithms themselves or to statistical noise.

The noisy-data scenario is extremely important in the web domain. The web is a huge repository of parallel documents that machine translation systems leverage for training data, and we continually extract content from noisy online sources. Unlike the above evaluation setup, we are concerned with scenarios where the data has a relatively high degree of noise, where by 'noise' we mean both non-perfect translations but also additional content on one side that is not translated at all. Both kinds of noise should be dealt with appropriately: the first introduces imperfect training data, while the second could eliminate good translations, or might send word alignment into a frenzy.

Because prior work mostly focused on the clean-data scenario, it is unknown whether previous evaluations would hold for noisy input. This makes it difficult to judge how these algorithms would compare to each other under more noisy conditions, or when any other experimental dimension is varied, such as domain and the language pair in question.

### 3.1 Creating Noisy Datasets for Evaluation Purposes

How can we create a noisy-data scenario under which to evaluate a sentence alignment algorithm? One approach is to mimic prior work: in a dataset that is known to be noisy, have an annotator select the sentence pairs that should be aligned to each other. However, this approach would be expensive

and time-consuming.

We propose a completely different approach. Rather than attempting to annotate corresponding sentences in a dataset that is known to be noisy, we deliberately introduce noise into a dataset that is already perfectly-aligned (and for which, as a consequence, we already know the sentence correspondence).

Specifically, we start with a parallel dataset $D$ that we know to be perfectly-aligned. Such datasets are abundant and readily available for MT researchers in the form of a myriad of tuning and test datasets across many language pairs and domains. We introduce noise into $D$ (using any of the methods described below and detailed in subsection 4.2) to obtain a modified dataset $D'$. The source side of $D'$ is a subset of the source side of $D$ (possibly reordered), and the same holds for the target side. Since we know what the correct sentence alignments are in $D$, we also know, by mere construction, what the correct alignments in $D'$ are as well. This allows us to easily compute precision and recall of a sentence alignment algorithm when it is given $D'$ as input, without the need to collect a single annotation.

We employ several methods to create a noisy dataset $D'$ from a perfectly-aligned dataset $D$:[2]

- **Clean dataset.** The source and target sides of $D'$ are exactly the unaltered source and target sides of $D$. This represents the easiest test set for a sentence aligner, as the test set consists entirely of 1-to-1 mappings, all of which fall exactly along the search matrix diagonal.

- **Random deletions.** The source side of $D'$ is a subset of the source side of $D$, where the number of discarded sentences is determined by a source deletion rate $del_s$. For example, for a dataset $D$ with 1000 sentences on the source side and $del_s = 0.10$, the source side of $D'$ consists of 900 randomly-chosen sentences from the source side of $D$ (with no reordering). The target side of $D'$ is created similarly, using a target deletion rate $del_t$. Note that the deletion on the target side is done independently from the deletion on the source side. That is, the probability of deleting the $i$th sentence on the target side is $del_t$, regardless of whether the $i$th sentence on the source side was deleted or not.

- **Random combinations.** The source and target sides of $D'$ are the same as those from $D$, but with random consecutive pairs of sentences combined into a single sentence. The degree to which sentences are combined is determined by source and target combination rates $comb_s$ and $comb_t$. For example, for a dataset $D$ with 1000 sentences on the source side and $comb_s = 0.10$, 100 sentence pairs (each consisting of consecutive sentences) are chosen randomly, and each pair is combined into a single sentence, yielding a set of 900 source sentences in $D'$. The goal of this scenario is to test the aligner's ability to recover 1-to-many and many-to-1 mappings, rather than focusing solely on 1-to-1 mappings.[3] As with random deletions, the combination processes on the source side and on the target side are independent from each other.

- **Randomized order.** The source side of $D'$ consists of the source side of $D$, but in random order. The target side of $D$ is also randomized.

- **Length-aligned from same dataset.** The source side of $D'$ is exactly the same as the source side of $D$. The noise is introduced into the target side, where all the target sentences from $D$ are preserved, but they are reordered. The reordering is not completely stochastic. Rather, an attempt is made to have the sentences length-aligned as much as possible. This is somewhat of an adversarial scenario, since a length-based alignment method would align too many sentences that are completely unrelated to each other.

- **Different datasets.** The dataset $D'$ is formed by taking two datasets $D_1$ and $D_2$, and aligning the source side of $D_1$ with the target side of $D_2$, and vice versa. A good sentence aligner would deem that the source and target sides are unrelated, yielding a very low alignment rate.

[2] In a few of our experiments, we make use of *two* datasets (that are non-overlapping and non-related), say $D_1$ and $D_2$, to create $D'$. The way we frame the creation of $D'$, as a mapping from a single dataset $D$, still applies here: $D$ is simply the concatenation of $D_1$ and $D_2$.

[3] With high enough combination rates, many-to-many mappings arise as well.

## 4 Experimental Results

Even though this paper is not mainly concerned with comparing aligners to each other, we utilize our proposed framework and apply it to three different aligners as a demonstration. In this section, we describe the aligners to be compared, and provide specific details about how our test sets were generated. We then describe the metrics we use, and present results based on these metrics.

### 4.1 Sentence Aligners

The first aligner (LEN) is a length-based aligner based on the algorithm described in Brown et al. (1991). It segments the source and target sides by finding the highest-likelihood segmentation according to a model describing the relationship between source sentence length and target sentence length. In particular, this relationship is modeled using a Poisson distribution that has as its mean the length ratio observed in the dataset to align.[4]

The second aligner (MRE) is based on Moore's algorithm (Moore, 2002), which makes use of the length-based aligner's output to build a tentative model 1. Moore's algorithm takes the output from this "first phase" and builds a bilingual lexicon that allows it to compute translation model scores. For a given pair of sentences, the likelihood that they are translations of each other is now computed based not only on their lengths, but also on their lexical similarity.

The third aligner (MRE+) is similar to the second aligner, but uses a much stronger translation model. The stronger translation model is simply the translation system that has already been built for that particular language pair and now helps aligning new data. While this requires the availability of external resources, this setup closely resembles the resources we have, given our parallel training datasets. We note here that our evaluation datasets have no overlap with the data used to train the translation models used by MRE+.

### 4.2 Noisy Dataset Generation

For random deletions, we use six different deletion rates (from 0.00 to 0.25, with 0.05 increments), both on the source side and the target side, for a total of 35 test sets. For random combinations, we use four different combination rates (from 0.00 to 0.15, with 0.05 increments), both on the source side and the target side, for a total of 15 test sets. Note that we do not consider the case when both deletion/combination rates are 0.00, since that mimics the clean-dataset scenario.

For the length-aligned scenario, we align each source sentence with a randomly-selected sentence from the target side that is closest in length to that source sentence. (We take the target-to-source length ratio into consideration, and multiply the source length by that ratio before trying to find the closest-length target sentence.) If several target sentences have lengths that are equally close to the desired length, we pick one at random.

We note here that if the source sentences are processed sequentially, there will be a clustering of overly long target sentences at the bottom of the dataset, since such sentences are never chosen based on length – they are simply too long. Therefore, we process the source sentences in random order rather than sequentially, to avoid this clustering of long sentences.

### 4.3 Performance Metrics

We report the following metrics for quantitatively evaluating and describing the output of the sentence aligner:

- **Precision**: of the sentence pairs produced by the aligner, what percentage are sentence pairs in the gold-standard dataset $D$?
- **Recall**: of the sentence pairs in the gold-standard dataset $D$, what percentage are produced by the aligner?
- **Alignment rate**: what proportion of the sentences in the input dataset $D'$ were aligned by the aligner? Due to the possibility that the source and target sides of $D'$ have different sizes, there are two alignment rates, and we report their average.[5]

Higher precision and higher recall are, by definition, indicators of better performance. This cannot be said of the alignment rate. For instance, consider the noisy deletion scenario of 3.1 above. By mere construction of $D'$, there will be source (resp. target) sentences that should not be aligned to anything on the target (resp. source) side, since we deliberately deleted the corresponding sentence. In such cases, an alignment rate of 100%

[4]Note that we follow Moore (2002) in using a Poisson distribution instead of the Gaussian of Brown et al.

[5]Of course, the dataset returned by the aligner always has source and target sides of equal sizes.

| Language Pair | Test Scenario | LEN | MRE | MRE+ |
|---|---|---|---|---|
| EN-ES | Clean (no noise) | 100%, 82%, 82% | 100%, 85%, 85% | 100%, 99%, 99% |
| | $del_s = del_t = 0.05$ | 100%, 46%, 44% | 99%, 71%, 68% | 100%, 96%, 91% |
| | $comb_s = comb_t = 0.05$ | 100%, 39%, 38% | 99%, 66%, 64% | 100%, 92%, 89% |
| | Randomized | 0%, 0%, 1% | 0%, 0%, 4% | 34%, 1%, 4% |
| | Length-aligned | 0%, 0%, 82% | 0%, 0%, 15% | 0%, 0%, 7% |
| EN-AR | Clean (no noise) | 100%, 55%, 55% | 100%, 60%, 60% | 100%, 89%, 89% |
| | $del_s = del_t = 0.05$ | 99%, 27%, 26% | 99%, 44%, 42% | 100%, 82%, 78% |
| | $comb_s = comb_t = 0.05$ | 99%, 22%, 21% | 99%, 41%, 39% | 99%, 77%, 74% |
| | Randomized | N/A, 0%, 0% | 17%, <1%, <1% | 26%, <1%, 1% |
| | Length-aligned | 0%, 0%, 59% | 0%, 0%, 9% | 5%, <1%, 2% |
| EN-CH | Clean (no noise) | 100%, 66%, 66% | 100%, 72%, 72% | 100%, 97%, 97% |
| | $del_s = del_t = 0.05$ | 100%, 40%, 39% | 99%, 56%, 55% | 100%, 92%, 88% |
| | $comb_s = comb_t = 0.05$ | 99%, 35%, 34% | 99%, 52%, 50% | 99%, 87%, 82% |
| | Randomized | 0%, 0%, <1% | 0%, 0%, <1% | 29%, <1%, 2% |
| | Length-aligned | 0%, 0%, 62% | 0%, 0%, 13% | 2%, <1%, 5% |
| Average (over the 3 LP's) | Clean (no noise) | 100%, 68%, 68% | 100%, 72%, 72% | 100%, 95%, 95% |
| | $del_s = del_t = 0.05$ | 100%, 38%, 36% | 99%, 57%, 55% | 100%, 90%, 86% |
| | $comb_s = comb_t = 0.05$ | 99%, 32%, 31% | 99%, 53%, 51% | 99%, 85%, 82% |
| | Randomized | 0%, 0%, <1% | 6%, <1%, 2% | 30%, 1%, 2% |
| | Length-aligned | 0%, 0%, 68% | 0%, 0%, 12% | 2%, <1%, 5% |

Table 1: Results of the comparative experiment of the three aligners. For brevity, we report the results for only five scenarios (per language pair and aligner) out of the more than fifty scenarios we propose. Each cell contains three percentages: precision, recall, and alignment rate. The N/A precision value for LEN in the EN-AR randomized scenario indicates the aligner produced no output.

for example (i.e. all input sentences were aligned to some other sentence) is indicative of pervasive alignment rather than good performance.[6]

Hence, alignment rate is not a performance measure in the conventional sense, as it is not an objective to be maximized or minimized. Still, it is a useful descriptor that sheds light on the aligner's behavior, as we see in the next subsection.

### 4.4 Results

We carried out experiments covering three language pairs: English-Spanish, English-Arabic, and English-Chinese. The comparative experiment is quite telling, and the results (Table 1) point to consistent and noticeable differences between the three examined aligners. While all aligners have very high alignment precision rates in non-adversary scenarios, always exceeding 99%, the difference is in how well they recover sentence pairs that should be aligned to each other, illustrated by significant differences in recall rates.

The clearest trend is that the length-based algorithm (LEN) performs worse than Moore's algorithm (MRE), which in turn benefits quite a bit when it's aided by an external strong translation model (MRE+). It is worth pointing out that the gap between MRE and MRE+ is typically larger than the gap between LEN and MRE, suggesting the important of external bilingual resources to aid the sentence aligner.

The results of the adversary scenarios (randomized and length-aligned) are particularly interesting. Looking at precision and recall alone, it might seem that there is not much to separate the three algorithms. For example, they all have 0% precision and 0% recall in the length-aligned EN-ES scenario (fifth row of Table 1). However, looking at the alignment rate, we find that LEN was prone to over-aligning the data, having an (unnecessarily very high) alignment rate of 82%. On the other hand, MRE and MRE+, have much lower alignment rates of 15% and 7%, respectively. This means that they would introduce only a fraction of the

[6]Even an oracle aligner with perfect precision and recall will almost surely have an alignment rate less than 100% (or even 90%) when $D'$ is constructed using high deletion rates.

bad data that LEN would, which is a great advantage for MRE and especially MRE+.

## 5 Applications of the Evaluation Framework

In the previous section, we utilized our framework to perform a comparison between three different aligners, by evaluating them under various noisy-data circumstances. In this section, we use our framework in two more applications relevant to sentence alignment and machine translation.

### 5.1 Fine-tuning Aligner Parameters

We explore using the evaluation setup to fine-tune the parameters of the MRE+ algorithm. Lacking a principled way to evaluate the aligner's output, it was not possible to fine-tune the aligner's various parameters. Now, equipped with our evaluation framework, it is possible to quantitatively determine the effect of changing the value of any parameter, and pick the best value. This is preferable to accepting whatever default parameters are in already place, which are more than likely suitable for a specific domain, dataset, or low-to-nonexistent noise.

#### 5.1.1 Experimental Design

We fine-tune the parameters of the MRE+ algorithm by optimizing its performance on a tuning dataset generated using the noisy deletion setup, and then measure its performance on a different evaluation set that was also generated using the noisy deletion setup. We investigate two cases, one with $del_s = del_t = 0.05$, and one with $del_s = del_t = 0.20$, to examine the benefit of fine-tuning both under a relatively low noise level and under a relatively high noise level.

We optimize the performance of the MRE+ algorithm along three dimensions:

- **Prior probabilities (PRIOR).** As explained in section 2, sentence alignment is essentially a segmentation of the source and target sides of the parallel dataset. In addition to relying on length similarity and lexical correspondence, the MRE+ aligner also relies on a set of prior probabilities for each insert/delete/align action it could take. By default, the probability assigned to deletion and insertion was set at 0.02. It is reasonable to assume that this might be too low, especially for highly-noisy input data, and so this is the first dimension that we optimize.

- **Search beam size (SIZE).** The algorithm also pays attention to the location of a candidate sentence pair. While positional similarity does not play a direct role in computing the alignment probability, the aligner does prune the search space based on location. For example, when considering a sentence half-way through the source side, only sentences that are close to the half-way point in the target side will be considered. How far the aligner is willing to deviate from the diagonal[7] is a tunable parameter, making it our second dimension.

- **Alignment threshold (THRESHOLD).** The aligner assigns a probability to each sentence pair it considers for alignment, reflecting its confidence that the sentence pair should be aligned. By default, the aligner eliminates any sentence pair that fails to meet a threshold of 0.99. This alignment threshold is the third dimension we optimize, as it should be lowered or increased to reflect our confidence in the translation model and/or the variability of the length-correspondence model.

#### 5.1.2 Experimental Results

The results in Tables 2 and 3 show the benefit of optimizing the aligner's parameters. It is beneficial to optimize the prior probabilities and the alignment threshold, as indicated by higher recall rates compared to the default values. On the other hand, the tuning of the search beam size had minimal impact. This indicates that the mistakes made by the sentence aligner are usually model errors rather than search error.

The effect of optimizing the prior probabilities is more pronounced in the high-noise scenario (Table 3), where it proves to provide the most gain over the baseline. Contrast this with the low-noise scenario (Table 2), where optimizing the alignment threshold is at least equally important, if not more so. This is to be expected, since the default prior of 0.02 in the high-noise scenario significantly underestimates the amount of deletion that has actually taken place, making the prior the most important parameter to optimize.

[7] If we were to create a grid of alignment probabilities, this pruning of the search space means that grid cells far off the *diagonal* of this grid are never considered.

| Tuned parameter(s) | EN-ES | EN-AR | EN-CH |
|---|---|---|---|
| None | 95.7% | 82.4% | 92.0% |
| PRIOR | 96.2% | 85.6% | 93.5% |
| SIZE | 95.8% | 82.8% | 92.0% |
| THRESHOLD | 96.8% | 86.7% | 92.9% |
| All | 97.1% | 87.5% | 93.7% |

Table 2: Results of the MRE+ fine-tuning experiment for the 0.05 deletion rate scenario. For clarity, we show only recall rates – all precision rates are 99% or higher.

| Tuned parameter(s) | EN-ES | EN-AR | EN-CH |
|---|---|---|---|
| None | 87.8% | 68.1% | 81.9% |
| PRIOR | 92.7% | 81.5% | 88.4% |
| SIZE | 88.0% | 68.8% | 82.3% |
| THRESHOLD | 89.3% | 70.4% | 84.3% |
| All | 93.0% | 82.8% | 90.6% |

Table 3: Results of the MRE+ fine-tuning experiment for the 0.20 deletion rate scenario. For clarity, we show only recall rates – all precision rates are 98% or higher.

It is worth pointing out the work of Yu et al. (2012), who perform a comparative study of sentence aligners, and show that Moore's algorithm does not perform as well as other aligners on a noisy dataset. As they provide no details regarding the values of the various parameters of Moore's algorithm, one can assume that they used default values and performed no tuning. Of course, such tuning would not have been easy to perform, given the lack of a tuning dataset. This is exactly why we propose our evaluation framework, so that future researchers would not have to guess parameter values or accept default values if they believe that would lead to suboptimal performance. Given the results of our experiments, it is conceivable that the performance of Moore's algorithm in Yu et al.'s work (and other algorithms they examined as well) might have been improved had their parameters been optimized.

## 5.2 Using Sentence Alignment to Filter Training Data

Much of our training data comes from noisy sources, both online and otherwise. Due to the vast amount of data, it is not possible to go through it to discard noisy sentence pairs. Now, equipped with a better understanding of our sentence aligner and its performance, we use it to trim down our training data by eliminating sentence pairs to which the aligner does not assign a high weight.

### 5.2.1 Experimental Design

We provide our current training data as input to the sentence aligner, and treat the output of the aligner as a filtered version of our data, since sentences that are discarded (not aligned) by the aligner tend to be noisy data. To evaluate the effectiveness of this process, we compare models trained with pre-filtered data vs. ones trained with the filtered data. We examine how the filtering affects the data and model size, since trimming those down would speed up training and translation. This is especially relevant for us given the large number of language pairs for which we train models. To ensure the translation quality doesn't degrade, we measure the effect on translation quality for two in-house evaluation datasets.

We consider three scenarios:

- **No filtering.** As a baseline, we use our training data as-is to train the MT system, without any filtering.
- **Uniform filtering.** We provide our training data as input to the sentence aligner, and use the aligner's output as the training data to train the MT system. (We refer to this as 'uniform' filtering in contrast to the next scenario.)
- **Filtering 'web' datasets.** Here, we apply sentence alignment filtering only to certain hand-picked datasets that we believe to contain a relatively high level of noise. The datasets are not picked by inspecting their content, but simply by deciding that any dataset that came from online sources (aka 'web' data) should undergo filtering.

### 5.2.2 Experimental Results

We performed our filtering experiments on two systems, Arabic-English and Urdu-English, with the results displayed in Tables 4 and 5, respectively. In all cases but one, the BLEU score went up or down by less than a quarter of a point, indicating general stability in performance quality.

This line of experiments is still in progress. We plan to carry out another set of experiments where

| Scenario | Data Size | Model Size | Test1 BLEU | Test2 BLEU |
|---|---|---|---|---|
| No filtering | 100% | 100% | 31.44 | 30.57 |
| All filtered | 94.8% | 96.7% | 31.29 | 30.34 |
| Web only | 96.6% | 96.0% | 31.54 | 30.52 |

Table 4: Results of the data filtering experiments for the Arabic-English system.

| Scenario | Data Size | Model Size | Test1 BLEU | Test2 BLEU |
|---|---|---|---|---|
| No filtering | 100% | 100% | 38.03 | 13.32 |
| All filtered | 81.6% | 85.9% | 38.19 | 13.13 |
| Web only | 99.1% | 99.1% | 37.80 | 12.78 |

Table 5: Results of the data filtering experiments for the Urdu-English system.

the prior deletion probability is customized for each portion of our training data, based on our belief of how noisy that portion of the dataset is. We are also expanding the experiments to include more language pairs.

## 6 Related Work

Singh and Husain (2005) evaluate several sentence alignment algorithms. Their work does have a hint of proposing a fuller evaluation framework, in that they have one test scenario where noise is added to their test set (in the form of adding sentences from another, unrelated dataset). Another major difference from our work is that they rely on manual evaluation of the output, as is the case for much of prior work.

Moore does point out that the error rates obtained by his algorithm are very low partly because the data being aligned is highly parallel, therefore making it "fairly easy data to align" (Moore (2002), p. 142). He therefore presents one additional experiment where a single block of sentences is deleted from one side of the input to mimic a noisy condition. While this is similar in spirit to our noisy deletions scenario, it introduces only a very small amount of noise in practice. This is because the deleted sentences are all sequential rather than being at different positions in the corpus, are all on one side of the corpus, and since the deletion rate was very low (varied up to only 3.0%). Case in point, the resulting dataset was still very easy to align, with error rates that remained below 2.0% even for the baseline aligner.

Yu et al. (2012) use the BAF dataset (Simard, 2006) as an evaluation dataset, since it is known to contain a relatively high degree of 0-1 and 1-0 beads (what they call "null links"), and use that dataset specifically to evaluate an alignment algorithm customized to handle noisy data. Similarly, Rosen (2005) evaluates several aligners using three datasets, one of which is characterized as being more noisy than the others.

Abdul-Rauf et al. (2012) compare several algorithms to each other, across several datasets, including the noisy BAF dataset. However, they do not propose a full framework for evaluating sentence alignment itself, and instead emphasize the differences in performance of MT systems trained on the aligned data.

There is a good amount of prior work dealing with filtering noisy data from parallel datasets. Taghipour et al. (2010) propose a discriminative framework to filter noisy sentence pairs from parallel data, and apply it to a Farsi-English dataset. Denkowski et al. (2012) briefly describe a filtering method to clean up training data for a French-English system submitted to WMT 2010, relying on deviations from typical values for certain statistical measures to identify noisy sentence pairs.

## 7 Conclusion

In this paper, we proposed a new evaluation framework for sentence aligners, which is specifically designed with noisy-data conditions in mind. Our approach is unique in that it requires absolutely no manual labeling, and relies on parallel datasets that are already in existence. We provide several methods to deliberately introduce noise into a dataset that is already perfectly-aligned, thus creating a whole host of evaluation test sets quickly and at no cost.

Our framework allows us and other researchers to easily compare and contrast several aligners to each other. Furthermore, our framework can be used to improve the performance of an aligner by facilitating the fine-tuning of any or all of its hyperparameters.

## References

Sadaf Abdul-Rauf, Mark Fishel, Patrik Lambert, Sandra Noubours, and Rico Sennrich. 2012. Extrinsic evaluation of sentence alignment systems. In *Proceedings of LREC Workshop on Creating Cross-*

*language Resources for Disconnected Languages and Styles, CREDISLAS*, pages 6–10.

Fabienne Braune and Alexander Fraser. 2010. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of COLING: Poster Volume*, pages 81–89.

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of ACL*, pages 169–176.

Stanley F. Chen. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of ACL*, pages 9–16.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The CMU-Avenue French-English translation system. In *Proceedings of the NAACL Workshop on Statistical Machine Translation*, pages 261–266.

William A. Gale and Kenneth W. Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of ACL*, pages 177–184.

Martin Kay and Martin Röscheisen. 1993. Text-translation alignment. *Computational Linguistics*, 19(1):121–142.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*, pages 79–86.

Philippe Langlais, Michel Simard, and Jean Véronis. 1998. Methods and practical issues in evaluating alignment techniques. In *ACL/COLING*, pages 711–717.

Xiaoyi Ma. 2006. Champollion: A robust parallel text sentence aligner. In *Proceedings of LREC*, pages 489–492.

I. Dan Melamed. 1997. A portable algorithm for mapping bitext correspondence. In *Proceedings of ACL*, pages 305–312.

Arul Menezes and Stephen D. Richardson. 2001. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *Proceedings of the ACL Workshop on Data-Driven Methods in Machine Translation*, pages 39–46.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In Stephen D. Richardson, editor, *AMTA 2002: From Research to Real Users*, pages 135–144. Springer Berlin Heidelberg.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

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

Alexandr Rosen. 2005. In search of the best method for sentence alignment in parallel texts. In *Proceedings of SLOVKO*.

André Santos. 2011. A survey on parallel corpora alignment. In *Proceedings of MI-Star*, pages 117–128.

Michel Simard and Pierre Plamondon. 1998. Bilingual sentence alignment: Balancing robustness and accuracy. *Machine Translation*, 13:59–80.

Michel Simard, George F. Foster, and Pierre Isabelle. 1993. Using cognates to align sentences in bilingual corpora. In *Proceedings of the Conference of the Centre for Advanced Studies on Collaborative Research: Distributed Computing - Volume 2*, pages 1071–1082.

Michel Simard. 2006. The BAF: A corpus of English-French bitext. In *Proceedings of LREC*, pages 489–494.

Anil Kumar Singh and Samar Husain. 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.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Proceedings of NAACL*, pages 403–411.

Kaveh Taghipour, Nasim Afhami, Shahram Khadivi, and Saeed Shiry. 2010. A discriminative approach to filter out noisy sentence pairs from bilingual corpora. In *Proceedings of International Symposium on Telecommunications*, pages 537–541.

Jörg Tiedemann. 2007. Improved sentence alignment for movie subtitles. In *Proceedings of Recent Advances in Natural Language Processing*.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of ACL*, pages 72–79.

Jean Véronis and Philippe Langlais. 2000. Evaluation of parallel text alignment systems: The ARCADE project. In Jean Véronis, editor, *Parallel Text Processing: Alignment and Use of Translation Corpora*, pages 369–388. Kluwer Academic Publishers.

Qian Yu, Aurélien Max, and François Yvon. 2012. Revisiting sentence alignment algorithms for alignment visualization and evaluation. In *Proceedings of the LREC Workshop on Building and Using Comparable Corpora*, pages 10–16.

Bing Zhao and Stephan Vogel. 2002. Adaptive parallel sentences mining from web bilingual news collection. In *IEEE International Conference on Data Mining*, pages 745–748.