JMaxAlign: A Maximum Entropy Parallel Sentence Alignment Tool

Joseph Max Kaufmann

`max.kaufmann@dac.us`

Abstract

Parallel corpora are an extremely useful tool in many natural language processing tasks, particularly statistical machine translation. Parallel corpora for certain language pairs, such as Spanish or French, are widely available, but for many language pairs, such as Bengali and Chinese, it is impossible to find parallel corpora. Several tools have been developed to automatically extract parallel data from non–parallel corpora, but they use language-specific techniques or require large amounts of training data. This paper demonstrates that maximum entropy classifiers can be used to detect parallel sentences between any language pairs with small amounts of training data. This paper is accompanied by JMaxAlign, a Java maxent classifier which can detect parallel sentences.

Keywords: Parallel Corpora, Comparable Corpora, Maximum Entropy Classifiers, Statistical Machine Translation.

1   Introduction

Parallel corpora, one text translated into multiple languages, are used in all types of multilingual research, especially in Machine Translation (MT) tasks. Statistical Machine Translation (SMT) systems analyze parallel corpora in two languages to learn the set of rules that govern translation between them. Since SMT systems use statistical methods, they require lots of training data to produce useful results Smith et al. [2010]. Unfortunately, parallel corpora are very difficult to produce. Creating parallel corpora requires humans who are proficient in both the source and target languages. These translators usually are compensated in some form, which means that generating parallel corpora is both expensive and time consuming. It is possible to buy corpora, but not for every language pair. The Linguistic Data Consortium, one the largest providers of parallel corpora, contains corpora for less than 20 languages pairs [1]. Because of the utility of parallel corpora, several tools have been created which allow the automatic extraction of parallel corpora from non–parallel corpora. However, these tools have been designed to extract parallel corpora from a specific language pair, such as English–Hungarian (Tóth et al. [2005]) or Arabic–English Munteanu and Marcu [2005]. This paper describes JMaxAlign, a Java Maximum Entropy Alignment tool. JMaxAlign builds upon previous state-of-the-art research by using maximum entropy classifiers to detect parallel sentences. By doing so, it avoids dependence on hard–coded linguistic features (such as lists of stop words or cognates). This independence makes it useful for generation of parallel-corpora for low-resource language pairs, such as Hindi-Bengali or Chinese-Tamil.

2   Finding Parallel Sentences

The bulk of previous research that aims to detect parallel sentences uses a combination of two techniques: length-based similarity and lexical similarity.Length based similarity was originally developed by Gale and Church [1993]. Length-based similarity is the idea that long sentences are likely to have long translations and short sentences are likely to have short translations. This method is very effective for discarding non-parallel sentences. This is useful because parallel corpora extraction is frequently performed on a large corpora which may only contain a small amount of parallel data. When Adafre and de Rijke [2006] attempted to extract a parallel Dutch–English corpora from Wikipedia, they were able to substantially decrease the number of candidate sentences by discarding sentence pairs that have drastically different length. But to actually detect parallel sentences, they used the second technique, lexical similarity.

Lexical similarity is the fraction of the words in one sentence of the source language that are semantically equivalent to words in the corresponding sentence of the target language. There are many different ways to compute lexical similarity, but all require a bilingual dictionary. Adafre and de Rijke [2006] compute it by using a bilingual dictionary, and looking at the words in the English sentence that have translations in the Dutch sentence. Tools such as Hunalign Tóth et al. [2005] and the Microsoft Bilingual Sentence aligner Moore [2002] automatically generate their bilingual dictionary from the corpora, using pruning heuristics to narrow possible word translations. Other tools, such as Ma [2006] actually create a weighted bilingual dictionary, which uses TF-IDF to make rare words more indicative then parallelism.

---

[1]http://www.ldc.upenn.edu/Membership/Agreements/memberannouncement.shtml

While there are many ways to combine these two similarity measurse, previous research has shown using them to generate features for a maximum entropy classifier is an effective method. While there are no publicly available sentence tools that use this approach, it has been researched in several papers (such as Smith et al. [2010] and Mohammadi and QasemAghaee [2010]) and been shown to be a powerful technique. Specifically, Smith et al. [2010] used maximum entropy classifiers to detect parallel sentences in English and Spanish, and Mohammadi and QasemAghaee [2010] used them in the context of an Arabic–English pair. While English and Spanish belong to similar language families, English and Arabic are unrelated.

2.1 Motivation

The fact that maximum entropy classifiers achieved reasonable success when classifying English–Spanish parallel sentences and English–Arabic parallel sentences was the main motivator for the creation of JMaxAlign. This fact suggested that maximum entropy classifiers could learn the appropriate weights of lexical and sentence–length similarity. This is important because not all tools assume that these features need to be learned for a language pair. Hunalign, one of the most well known sentence alignment tools, does not make this assumption. Tóth et al. [2005] claim that "The relative weight of the two components [lexical and sentence–length similarity] was set so as to maximize precision on the Hungarian—English training corpus, but seems a sensible choice for other languages as well." Unfortunately, this is not the case. This is because sentences in agglutinative languages with rich morphologies (such as Hungarian or Turkish) generally have fewer words than semantically equivalent sentences in isolating languages (such as Spanish and English). This means that Hunalign's sentence–length measurement is *not* applicable to every language pair. Hunalign's implementation of lexical similarity suffers from two flaws. First, Hunalign attempts to assign a lesser weight to stop words (function words that primarily serve a grammatical purpose) when computing lexical similarity. This is logical, as stop words are not nearly as indicative of parallelism as rarer words. However, Hunalign uses hard–coded lists of English and Hungarian stop words, and provides no way to substitute a new list for different language pairs. Secondly, it does not account for why a word in one sentence is not aligned to a word in another sentence. A word can be unaligned because the dictionary does not include that word, or it can be unaligned because it truly has no corresponding alignment. The first means that the aligner lacks the knowledge necessary to compute the alignment, while the second is really indicates that the words are unaligned.
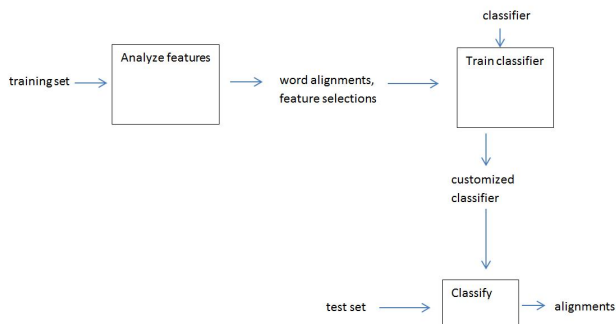
3 JMaxAlign

3.1 Architecture

JMaxAlign takes two parallel corpora as input. From these corpora, JMaxAlign computes build a probabilistic bilingual dictionary between the two languages of the corpora, and uses that dictionary to compute feature sets for each pair of parallel sentences[2]. These feature sets are then used to train a maximum entropy classifier for that language pair. The test data is then passed into the classifier, which outputs a boolean value for each sentence pair, indicating whether they are aligned or not. Figure 1 represents this process.

---

[2]Thanks to Stanford, for making the Stanford Classifier which is used as the maximum entropy classifier http://nlp.stanford.edu/software/classifier.shtml

Figure 1: Architecture of JMaxAlign

3.1.1 Lexical Alignments

After parallel corpora for training have been gathered, JMaxAlign computes the lexical alignments for each of the words in the parallel corpora. A lexical alignment is simply a set of possible translations (and optionally the probability of those translations) a word can have. Multiple words in one language can map to a single word in another. This is frequently due to differing uses of case. For example, German marks a noun and its modifier to indicate its case, while English does not. The English phrase *the dog* can be translated into German as der hund, den hund, dem hund, or des hundes.

These alignments are extremely important because they are the basis of the probabilistic bilingual dictionary that allows JMaxAlign to compute the lexical similarity of two sentences. JMaxAlign uses them to calculate the percentage of words in one sentence that have translations in the corresponding sentence. Turning parallel sentences into lexical alignments is a difficult task, but it has been extensively studied in the NLP community. JMaxAlign uses the Berkeley Aligner, an open–source tool written in Java that generates probabilistic word alignments from a set of parallel texts (the same texts that serves as the training data for the aligner). This aligner, described by Liang et al. [2006], works with either supervised or unsupervised training data (in this context, supervised means parallel sentences with hand–generated alignments, not just parallel sentences).

3.1.2 Feature Selection

Feature selection is the most important part of building a classifier. If a classifier is looking at the wrong features to decide whether sentences are parallel or not, it will not obtain accurate results.

Length Ratio The ratio of the length between the two sentences. Gale and Church [1993]

showed that the length ratio between sentences that are in fact parallel is an effective way to determine parallelism.

Percentage of Unaligned Words This measures the lexical similarity of the two sentences. Having more words unaligned means that the two sentences have a lower lexical similarity. JMAxAlign computes lexical similarity by taking the number of unaligned words in both sentences, and dividing it by the total number of words in both sentences.

Percentage of Unknown and Unaligned Words One of the weaknesses of Hunalign was that it did not account for words that were not aligned because they had never been seen in training. This feature helps the classifier account for the fact that when there is a low amount of training data sentences may appear not to be aligned due to the large percentage of unknown words. It is computed by taking the number of unaligned and unknown words in both sentences, and dividing it by the total number of words in both sentences.

Sum of Fertilities Brown et al. [1993] defines the fertility of a word as the number of words it is connected to. One word can be aligned to multiple words in its corresponding translation. As Munteanu and Marcu [2005] writes, "The presence in an automatically computed alignment between a pair of sentences of words of high fertility is indicative of non–parallelism". A typical example of this is stop words. The appearance of words such as *a*, *the*, or *an* do not help us decide parallelism nearly as much as the appearance of words such as *carpet*, *burning*. or *elephant*. The latter set of words is much rarer, and therefore much more indicative of parallelism. In Hunalign, lists of hard–coded Hungarian and English stop words were used. Since it is time–consuming to create lists of stop words for each language pair, the sum fertilities measure is used.

Longest Contiguous Span Long contiguous spans of aligned words are highly indicative of parallelism, especially in short sentences. JMaxAlign compute longest contiguous span of aligned words in each sentence of the pair, and pick the longer of the two.

Alignment Score Following Munteanu and Marcu [2005] The alignment score is defined as as the normalized product of the translation probabilities of aligned word pairs. This is indicative of parallelism because non–parallel sentences will have less alignments and a lower score.

3.1.3   Classifier Training

After computing features from the training and testing data, the next step is to train a maximum entropy classifier. Even though JMaxAlign only requires the user to put in parallel sentences (positive training data), maximum entropy classifiers require negative training (non–parallel sentences). JMaxAlign simply randomly pairs sentences from the parallel corpus together to create negative training examples. When designing their Arabic–English maximum entropy classifier, Munteanu and Marcu [2005] assumed that randomly-generated nonparallel sentences were not sufficiently discriminatory. They created negative training examples by the same method that JMaxAlign did, but they filtered the sentences to only include the subset that had a length ratio less than 2.0, and over 50% of the words aligned. Munteanu and Marcu [2005] claimed that sentences that did not meet these characteristics

could immediately be classified as non–parallel, and so there was no need to train the classifier on these sentences. This claim is problematic because it assumes that those numbers are appropriate choices, similar to the way that Tóth et al. [2005] assumed that their length–ratio was appropriate for all values. One of the main benefits of maximum entropy classifiers is that it does not force assumptions about how a certain set of linguistic properties will hold constant regardless of language pairs. Observe the following Turkish and English sentences

Turkish: ekoslovakyallatramadklarmzdan mydnz? *English*: Were you one of those who we failed to assimilate as a Czechoslovakian?

This is an extreme example, but it illustrates the dangers of assuming linguistic constants. These sentences are parallel, but would be immediately discarded by Munteanu and Marcu [2005]'s filter. Randomly generating negative sentences further benefits JMaxAlign by allowing it to learn the importance of the other features. While the features of the filter have been shown to be extremely useful discriminators, there is no need to include that bias in the classifier. There may be sentences that don't pass the filter but provide useful information about how fertilities or contiguous spans contribute to parallelism. Discarding them causes an unnecessary loss of information.

4 Results

4.1 Linguistic Similarity

Detecting similar sentence pairs is easier when the languages are more linguistically related. For example, when detecting English-Spanish language pairs, cognates could be used in the absence of a bilingual dictionary, but this is not the case for Arabic–Chinese. To evaluate JMaxAlign on its sensitivity to the language family of the languages, it was tested on two types of language pairs: language pairs from the same family ("similar language pairs"), and language pairs from different families ("dissimilar language pairs") In the first test, the training data used included 2000 parallel and 1000 non-parallel training examples, and the testing data included 5000 parallel examples and 5000 non-parallel examples. The parallel data was obtained from the Open Subtitles Corpora, a collection parallel sentences obtained from aligning movie subtitles Tiedemann et al. [2004]. The results are in Table 1 and Table 2.

Table 1: Similar Language Pairs

Language 1	L1 Family	Language 2	L2 Family	Precision	Recall	F–Score
English	Germanic	German	Germanic	86.75	87.62	87.81
Spanish	Italic	Italian	Italic	87.57	85.49	86.51
Estonian	Uralic	Hungarian	Uralic	54.99	99.75	70.89
Polish	Slavic	Russian	Slavic	90.08	92.51	91.28

Table 1 and 2 show that JMaxAlign is somewhat sensitive to the language families to which the languages belong. The F–Scores for the similar language pairs are, in general, higher than those for the dissimilar pairs. However, JMaxAlign is still able to produce useful results (F–Scores between 75% and 80%) for languages that are very dissimilar. The high recall scores that JMaxAlign achieves indicates that it is very good at classifying non-parallel

Table 2: Dissimilar Language Pairs

Language 1	L1 Family	Language 2	L2 Family	Precision	Recall	F–Score
Arabic	Semitic	Spanish	Italic	69.1	83.85	75.76
German	Germanic	Italian	Italic	83.89	80.3	82.04
Spanish	Italic	Russian	Slavic	79.84	79.17	79.50
Hungarian	Uralic	Russian	Slavic	87.05	74.09	83.09

sentences as non-parallel without the help of a filter, like the one Munteanu and Marcu [2005] proposes. Not using a filter is a non-trivial benefit for two reasons. First, it does not limit the amount of negative sentences that can be generated. If a training corpus only contains 5000 sentences, it may not be possible to find enough sentences that are not parallel and have a length ratio less than 2.0, and have over 50% of the words aligned.

4.2   Domain Sensitivity

No research has been done regarding the effect of domain on maximum entropy classifiers that generate parallel corpora, but the effect of domain is very important to any type of classifier. It is especially important for the use case of JMaxAlign, since it is very possible that the comparable corpora from which parallel corpora are being extracted may be in a different domain than the training data. For example, someone wishing to extract parallel corpora from Wikipedia might be forced to train JMaxAlign on a corpora constructed from government documents. JMaxAlign classifies the test sentences based on the similarity between their features and the features of the training data. If the corpora are from different domains, they will have different features. For this reason, training and testing on corpora from different domains can lower the quality of the results in many NLP tasks. This is particularly relevant to sentence alignment because extraction from a new data source may require training on out–of– domain parallel corpora.

To test the sensitivity of JMaxAlign to domain, two corpora were chosen from the Open Source Parallel Corpora[3], an online collection of many corpora. In both of these corpora, sentences were already segmented, so sentence-aligned gold standard data was available. The first is the KDE corpora, a collection of localization files for the K Desktop Environment. The second is the Open Subtitles corpora, which was generated aligning movie subtitles in different languages based on the time at which they appeared on the screen. Both corpora already marked sentence boundaries, so no sentence splitting tools were needed.

These corpora exhibit several differences. First, they are not equally parallel. Tiedemann et al. [2004] makes no observation about the accuracy of the Open Subtitles Corpora, but they note that "not all translations are completed and, therefore, the KDE corpus is not entirely parallel". So there may be some noise in the training data. Second, they both capture a different type of language data. The Open Subtitles corpora are a collection of transcriptions of spoken sentences, while the KDE corpora are a collection of phrases used to internationalize the KDE.

The KDE corpora contain a very specific set of vocabulary. The data are also much more terse, and sentences are usually between one and five words. Conversely, the Open Subtitles

---

[3]http://opus.lingfil.uu.se/

corpora contain full grammatical sentences that are much longer. Since sentence-length similarity is a good indicator of non-parallelism, it was expected that cross-domain tests will show lower recall due to the different types of sentences in each of these corpora. This is because dictionaries generated by training on one set of corpora might not have large vocabulary overlap with a different corpora.

Tables 3 and 4 show the results of training and testing on all possible combinations of the KDE and Subtitles corpora. The "training" column indicates which corpora the training data came from and the "testing" column indicates which corpora the testing data came from. Each test was run with 5000 parallel and 3000 non-parallel training sentences, and tested on 5000 parallel and 5000 non-parallel sentences. In order to prevent the linguistic similarity of the language pairs could confound the effects of changing the testing and training domain, all domain combinations were tested on both similar and dissimilar language pairs. Table 3 shows the results of the domain tests for similar language pairs, and Table 4 shows the results for dissimilar language pairs.

Table 3: Similar Language Pairs

Language Pair	Precision	Recall	F–Score	Training	Testing
Spanish–French	96.26	49.04	64.93	Subtitles	KDE
Spanish–French	93.54	100.00	96.77	Subtitles	Subtitles
Spanish–French	99.9	49.97	66.62	KDE	Subtitles
Spanish–French	99.82	100.0	99.91	KDE	KDE
Polish–Russian	93.56	48.33	63.74	Subtitles	KDE
Polish–Russian	89.18	100	94.28	Subtitles	Subtitles
Polish–Russian	91.66	47.82	62.85	KDE	Subtitles
Polish–Russian	95.94	100.00	97.92	KDE	KDE

Table 4: Dissimilar Language Pairs

Language Pair	Precision	Recall	F–Score	Training	Testing
English–Japanese	94.84	48.67	64.33	Subtitles	KDE
English–Japanese	89.18	100.00	94.28	Subtitles	Subtitles
English–Japanese	87.4	46.63	60.82	KDE	Subtitles
English–Japanese	87.92	100.00	93.57	KDE	KDE
Arabic–Spanish	86.46	46.28	60.21	Subtitles	KDE
Arabic–Spanish	88.31	100	93.85	Subtitles	Subtitles
Arabic–Spanish	91.08	47.66	62.58	KDE	Subtitles
Arabic–Spanish	76.18	100.00	86.47	KDE	KDE

It is clear that altering testing and training domain has an effect on the quality of the results. Overall, the F–scores are higher for the similar language pairs when the testing and training domain are the same. This effect seems to be amplified on the dissimilar language pairs. The biggest effect of changing domain seems to e a significant drop in recall. This is probably due to the importance of the sentence-length feature. As stated earlier, sentence–length is extremely useful for determining which sentences are not parallel. By

training on the Subtitles corpora, the classifier learns that the length of the sentences is an important discriminator. But that assumption does not hold true for the KDE corpora, since the sentences are frequently not complete grammatical sentences. If the two corpora were not such drastically different registers of language, this effect might not be so dramatic.

4.3   Training Data Size

Since parallel corpora are rarely available, the fact that JMaxAlign requires parallel training data is disadvantageous. To study the effects of training data, we varied the size of training data used to build a Dutch–Chinese parallel corpora. Dutch–Chinese was chosen because the two languages are very dissimilar. If the chosen language pair contained two similar languages, it would be possible to achieve high F-Scores with very little training data. The previous results have shown that dissimilar pairs generally have lower F–Scores, so there is more room for growth, which allows us to better see the effects of varying training data. The values in the Table 5 were generated by testing on 5000 sentences, 2500 parallel and 2500 non-parallel. All data in these tests were taken from the Open Subtitles Corpora. For this experiment, balanced training data sets were used (i.e., the same number of parallel and non-parallel sentences were used in testing). Table 5 shows that adding more training data

Table 5: Effects of increasing training data

Training Data	Precision	Recall	F–Score
1000	85.44	94.48	82.59
2000	73.41	94.49	82.62
3000	73.44	94.49	82.64
4000	73.44	94.49	82.64
5000	73.44	94.49	82.64
10000	73.44	94.49	82.64

has little effect on the functionality of the classifier. The high recall in Table 5 indicates that JMaxAlign is good at detecting non-parallel sentences. However, the low precision indicates that it does not always accurately classifying parallel sentences. However, the F–Score at 2000 sentences is almost the same as the F-Score at 1000 sentences. This seems to indicate that JMaxAlign is not able to take advantage of extra data. But this is not the case. JMaxAlign can take advantage of extra data in unbalanced training sets. Munteanu and Marcu [2005]'s Arabic–English classifier achieved the best results when training on corpora that contained 70% parallel sentences and 30% non–parallel sentences.Table 6 is the result of training JMaxAlign on the same Dutch-Chinese corpora with various unbalanced training sets.

Table 6 shows that the balance of parallel and non-parallel sentences can be altered to improve performance. When there are 3000 parallel sentences and 1000 non-parallel sentences, JMaxAlign is able to achieve a much higher precision than that 1000 parallel sentences and 3000 non-parallel sentences. This is impressive, because despite the high ratio of parallel to non-parallel sentences, JMaxAlign is still able to do a good job classifying non-parallel sentences. When the number of parallel sentences is increased to 5000, and the number of non-parallel sentences to 2500, recall increases by 20%. However, this

Table 6: Training on Unbalanced Corpora

Parallel Sentences	Non-parallel Sentences	Precision	Recall	F–Score
1000	3000	51.20	95.95	66.77
3000	1000	87.57	73.81	80.10
2500	5000	63.76	93.79	75.91
5000	2500	77.02	94.81	84.9
5000	10000	64.08	93.83	76.14
10000	5000	80.24	95.10	87.02
20000	10000	76.08	95.64	84.74

causes the precision to drop by 10%, because the lower ratio decreases JMaxAlign's bias towards finding parallel sentences. Doubling the data to 10000 paralle l sentences and 5000 non-parallel sentences cause the precision to rise by 3%, somewhat negating this effect. But again doubling the amount of data (to 20,000 parallel sentences and 10,000 parallel sentences) causes the precision to drop again, possibly due to overfitting. From Table 6, it seems that the best results for JMaxAlign are achieved with between 10,000-15,000 parallel sentences, and 2,500-5,000 non-parallel sentences.

5   Conclusion

This work has resulted in several contributions. First, is has advanced the state–of–the art for parallel sentence detection. It has shown that maximum entropy classifiers, while effected by linguistic similarity and domain, can produce useful results for almost any language pair. This will allow the creation of parallel corpora for many new languages. If seed corpora that are similar to the corpora from which you want to extract parallel sentences can be made, JMaxAlign should prove a very useful tool. Even if seed corpora cannot be created, the KDE corpora can be used. JMaxAlign can be tuned for precision or recall by altering the amount of negative and positive training examples, so researchers can decide what trade–off is most appropriate for their task. JMaxAlign is also open–source, meaning that other researchers can modify it by changing the underlying classifier, or adding language–specific features. The code can be found at https://code.google.com/p/jmaxalign/.

References

S.F. Adafre and M. de Rijke. Finding similar sentences across multiple languages in Wikipedia. In *Proceedings of the workshop on new text: Wikis and blogs and other dynamic text sources*, pages 62–69, 2006.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311, 1993.

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

Percy Liang, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 104–111, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1220835.1220849. URL `http://dx.doi.org/10.3115/1220835.1220849`.

Xiaoyi Ma. Champollion: A robust parallel text sentence aligner. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC)*, 2006.

M. Mohammadi and N. QasemAghaee. Building bilingual parallel corpora based on wikipedia. In *International Conference on Computer Engineering and Applications (IC-CEA)*, volume 2, pages 264–268, 2010.

Robert C. Moore. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, AMTA '02, pages 135–144, London, UK, 2002. Springer-Verlag. ISBN 3-540-44282-0. URL `http://dl.acm.org/citation.cfm?id=648181.749407`.

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

J.R. Smith, C. Quirk, and K. Toutanova. Extracting parallel sentences from comparable corpora using document level alignment. In *NAACL-HLT*, pages 403–411, 2010.

Jörg Tiedemann, Lars Nygaard, and Tekstlaboratoriet Hf. The opus corpus – parallel and free. In *In Proceeding of the 4th International Conference on Language Resources and Evaluation (LREC*, pages 1183–1186, 2004.

Krisztina Toth, Richard Farkas, and Andras Kocsor. *Parallel corpora for medium density languages*, pages 590–596. Benjamins, 2005. URL `http://www.kornai.com/Papers/ranlp05parallel.pdf`.