SentAlign: Accurate and Scalable Sentence Alignment

Steinþór Steingrímsson[1], **Hrafn Loftsson**[1] **and Andy Way**[2]
[1]Department of Computer Science, Reykjavik University, Iceland
[2]ADAPT Centre, School of Computing, Dublin City University, Ireland
steinthor18@ru.is, hrafn@ru.is, andy.way@adaptcentre.ie

Abstract

We present SentAlign, an accurate sentence alignment tool designed to handle very large parallel document pairs. Given user-defined parameters, the alignment algorithm evaluates all possible alignment paths in fairly large documents of thousands of sentences and uses a divide-and-conquer approach to align documents containing tens of thousands of sentences. The scoring function is based on LaBSE bilingual sentence representations. SentAlign outperforms five other sentence alignment tools when evaluated on two different evaluation sets, German–French and English–Icelandic, and on a downstream machine translation task.

1 Introduction

Sentence alignment is the task of finding matching sentences in two parallel documents, as illustrated in Figure 1. It can be seen as a path-finding problem, with a list of source sentences on one axis in a two-dimensional graph and the target sentences on the other, as demonstrated in Figure 2. Each potential sentence pair is represented by a node in the graph, or nodes when multiple sentences are grouped together. The nodes are assigned values using a scoring function. The objective of the sentence alignment algorithm is to find the optimal path through the graph. Typically, the path is continuous, although gaps may occur when one of the documents has sentences without corresponding counterparts in the other document. The alignments can also be non-monotonous, where sentences cross, resulting in differences in sentence order between languages. This problem is often solved by chunking multiple sentences.

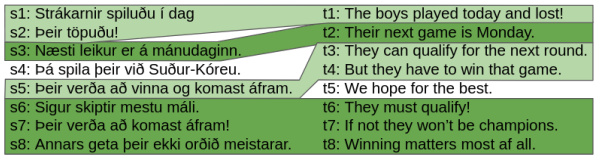


Figure 1: An automatic sentence alignment system aims to align source sentences $s_1, ..., s_n$ with target sentences $t_1, ..., t_n$ while using as few sentences as possible for each alignment. The figure shows examples of six alignment functions being applied while aligning eight sentences in Icelandic with eight sentences in English: Contraction (n–1), expansion (1–n), deletion (1–0), insertion (0–1), substitution (1–1) and merging (n–m).

Sentence alignment is a necessary processing step for parallel corpora to be useful for machine translation (MT). Neural machine translation (NMT) has been shown to be sensitive to misaligned training data (e.g. Khayrallah and Koehn (2018)) so an accurate sentence aligner is highly important for NMT to unleash the full potential of the parallel corpora it is trained on.

In this paper, we present SentAlign,[1] a sentence

[1]https://github.com/steinst/sentalign/

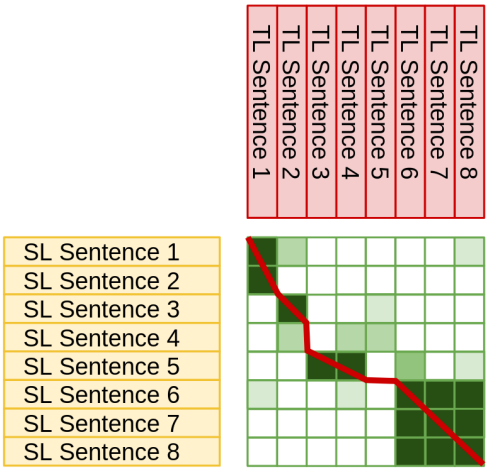


Figure 2: A two-dimensional alignment graph. The figure shows the path found through the graph after evaluating semantic similarity of all possible source (SL) and target language (TL) sentence pairs. Dark green nodes stand for the alignments selected by the system.

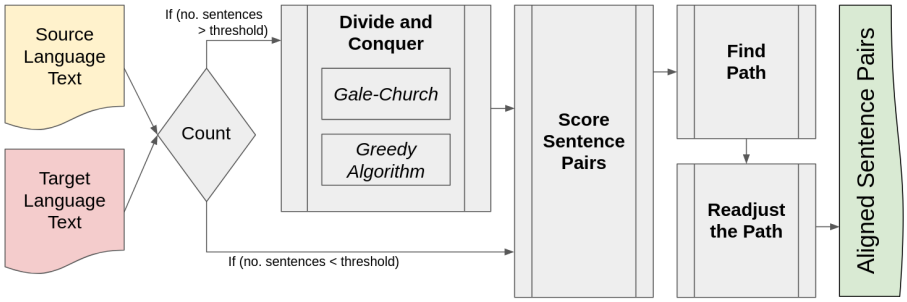


Figure 3: SentAlign Architecture.

aligner with a user-friendly command line interface, able to align very large documents. As shown in Section 4 it outperforms other available sentence aligners when evaluated on a common evaluation set, as well as on a downstream MT task. SentAlign evaluates all possible alignment paths in fairly large documents, with up to a few thousand sentences in each language, and activates a divide-and-conquer (DaC) approach to reduce running time when the number of sentences exceed a user-defined threshold. To identify matching sentences in two languages, SentAlign applies a scoring mechanism based on LaBSE (Feng et al., 2022), a model trained and optimized to produce similar representations for bilingual sentence pairs. The model, which employs both a masked language model (Devlin et al., 2019) and a translation language model (Conneau and Lample, 2019), is pretrained on monolingual and bilingual data in 109 languages.

2 Related Work

Gale and Church (1991) found that "the correlation between the length of a paragraph in characters and the length of its translation was extremely high". Motivated by that, they describe a method for aligning sentences based on a simple statistical model of character lengths.

The similarity score for Hunalign (Varga et al., 2005) has two main components: token-based and length-based. The token-based component searches for shared words in the two sentences, using an automatically generated lexicon or an external one. The length-based component is based on the ratio of longer to shorter sentences. The similarity score is calculated for every sentence pair in the neighbourhood of the diagonal of the alignment graph. Finally, a post-processing step iteratively merges 1–n ($n > 1$) and 0–1 segments wherever the resulting new segment has a better character-length ratio than the starting one.

Gargantua (Braune and Fraser, 2010) uses a two-step clustering approach to sentence alignment. It aims to find 1–n and n–1 alignments, but does not search for many-to-many alignments. It uses sentence length-based statistics considering relative lengths in comparison to the mean length of source and target sentences, and translation likelihoods of each target word with all source words, according to IBM Model-1 (Brown et al., 1990). It starts by looking for optimal alignments through the alignment matrix consisting only of 0–1, 1–0 and 1–1 correspondences. In a second step, the previously acquired alignments are merged into clusters containing up to R sentences (4 by default) on either the source or target size, and if the merge produces a better score it is accepted. The final alignments are found when an optimal score has been obtained for the whole graph.

Bleualign (Sennrich and Volk, 2010, 2011) uses MT and BLEU (Papineni et al., 2002) to align sentences. Even though BLEU has been criticised as a measure of translation quality and is not considered reliable on a sentence level (Callison-Burch et al., 2006), the authors of Bleualign point out that judging the quality of a translation is harder than deciding whether two sentences are possible translations of each other. Furthermore, they find that BLEU is very sensitive to misalignments, indicating that it should be capable of discriminating between aligned and unaligned sentence pairs. BLEU is usually measured on up to 4-grams. Too often, for the purposes of sentence alignment, this yields a score of 0 so Bleualign uses 2-grams. Furthermore, when comparing two sentences, the BLEU scores are different depending on which of the sentences is the hypothesis, due to the brevity penalty in BLEU. Therefore, Bleualign translates both directions when possible and uses the mean as the final score. In the first pass of the alignment algo-

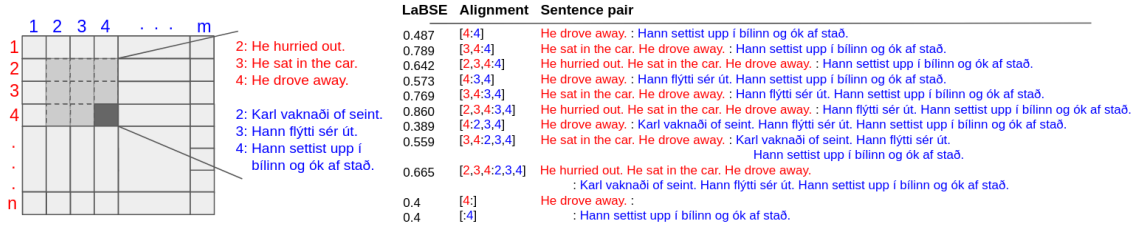


Figure 4: SentAlign searches for the best alignment that ends in node [4:4], with a maximum of 3 sentences merging on either side. LaBSE score is calculated for each alignment candidate. For insertions and deletions, where a sentence on either side is discarded, we assign the minimum threshold score, S_{min}.

rithm, a set of 1–1 beads are identified. In the second pass, all unaligned sentences that fall between the beads, are extracted and a list generated of all possible 1-, 2- or 3-sentence sequences composed of the unaligned sentences and the beads. BLEU scores are then calculated for the Cartesian product of the two lists. If any 1–n alignment scores higher than the bead, it is replaced in the graph and the step is repeated.

In Vecalign, Thompson and Koehn (2019) use the similarity between sentence embeddings as the scoring function, employing LASER (Artetxe and Schwenk, 2019) for scoring alignment candidates. In the alignment algorithm, recursive approximation is used to reduce the search space.

3 The SentAlign System

In this section, we present SentAlign, a highly accurate sentence aligner capable of evaluating all possible alignment paths through fairly large documents, using a LaBSE-based scoring mechanism. Our alignment approach is of quadratic complexity, $O(n^2)$, and in order to handle very large files, we apply a DaC approach. When the total nodes in the alignment graph exceed a user-defined maximum, by default set to $4,000,000$, the DaC-mechanism is activated in order to reduce the time complexity when aligning the documents.

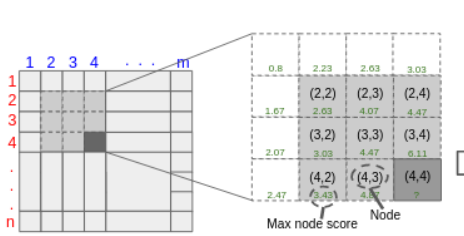
The main components of the SentAlign system illustrated in Figure 3 are the scoring mechanism, the alignment or pathfinding algorithm, a DaC-module to deal with very large files, and a readjustment module to compensate for shortcomings in the scoring mechanism.

3.1 Scoring

SentAlign uses LaBSE to score sentence-pair candidates. A minimum threshold score, defined by the user, is required for a sentence pair to be accepted. For each node $[i : j]$ in the alignment graph (where i is a sentence in the source language and j is a sentence in the target language), scores for all possible alignment combinations ending in that node are calculated. The user can set a maximum number of sentences that can be merged on either side of the alignment. If merging up to three sentences on each side is allowed, a total of $3 \times 3 = 9$ scores are compared for each node, as illustrated in Figure 4. If no alignment reaches the LaBSE threshold score, S_{min}, insertion and deletion functions are applied and the edges to the node obtain the score S_{min}. If the user wants to penalize long sentences, a user-defined maximum can be set for the number of words in either language. When either side of an alignment exceeds that maximum, a penalty is applied to the alignment score. The user can also define a maximum number of segment merges before a penalty is applied. That penalty is only applied in the pathfinding-phase (Section 3.2) and not when readjusting the path (Section 3.4). This penalty is set in order to favour shorter alignments and to deter the aligner from merging multiple sentences in one alignment when it is possible to find multiple shorter alignments instead. SentAlign seeks a maximum score for a given node in the alignment graph, S_{node}, and finds it by adding the alignment scores to the score of the node they connect from after penalties are applied.

3.2 Pathfinding

The alignment problem can be seen as a way of finding the optimal path through an $N \times M$ matrix, where N and M are the number of source and target sentences, respectively. As we allow for insertions, deletions and merges of multiple sentences on either side, we calculate the best path from the initial node $[0, 0]$ to all other nodes in the graph using a version of Dijkstra's algorithm (Dijkstra, 1959). Our objective is to maximize the score at each node, in contrast to the original algorithm,



Alignment	Score	Penalty	From node	Max node score	
[4:4]	2 x 0.49	0	(3,3)	5.45	
[3,4:4]	3 x 0.79	0	(2,3)	6.44	
[2,3,4:4]	4 x 0.64	0	(1,3)	5.19	
[4:3,4]	3 x 0.57	0	(3,2)	4.74	
[3,4:3,4]	4 x 0.77	0	(2,2)	5.71	
[2,3,4:3,4]	5 x 0.86	0	(1,2)	6.53	Best
[4:2,3,4]	4 x 0.39	0.02	(3,1)	3.61	
[3,4:2,3,4]	5 x 0.56	0.02	(2,1)	4.45	
[2,3,4:2,3,4]	6 x 0.67	0.02	(1,1)	4.80	
[4:]	1 x 0.40	0	(3,4)	6.51	
[:4]	1 x 0.40	0	(4,3)	5.27	

Figure 5: The maximum node score is calculated by adding the alignment score to the previously calculated maximum score of the node the alignment leads from. The LaBSE score is multiplied by the number of sentences comprising the alignment, e.g. alignment [2,3,4:3,4] has five sentences and thus the LaBSE score is multiplied by five. The max score for the node is found by adding the maximum score for node (1,2) to the alignment score.

which minimizes scores. This allows for large missing parts of text in either language without straying from the right path.

After all possible alignment scores have been calculated for a given node, an alignment function is chosen. If none of the alignments reach S_{min}, insertion and deletion alignment functions are applied and S_{min} assigned to the value of the resulting null alignments. If one or more of the possible n–m ($n \geq 1$) alignments has a score above the S_{min} threshold, we assign the alignment edge a value equalling the LaBSE score multiplied by the total number of sentences merged in both languages, and add penalty-adjustments to calculate the alignment score, as illustrated in Figure 5. Finally, we select the alignment obtaining in the highest score for S_{node}. This process is repeated for each node until node (n, m) is reached. At that point, we have the optimal score from $(0, 0)$ to (n, m) and mark the path by tracing backwards through the recorded edges.

3.3 Divide and Conquer

With more lines to align, the search space grows exponentially, affecting alignment speed. Zhang (2022) shows that for a quadratic time complexity sentence-alignment algorithm, chunking the parallel texts to be aligned using hard delimiters can reduce the time complexity to $O(n \log n)$. SentAlign allows the user to define a threshold for dividing up the search space. If the search space is larger than the user-defined threshold allows, the tool searches for high-confidence alignments to use as hard delimiters for dividing the search space into multiple smaller chunks, $k + 1$ chunks for k hard delimiters. The aim is to find the minimum amount of alignments to use as hard delimiters to split the parallel texts into chunks of manageable size.

SentAlign looks for 1–1 alignments in the middle half of the parallel texts to use as hard delimiters, with the middle half defined as the sentences in between the first and last 25% of the sentences in the texts. One of two approaches is chosen, depending on the size of the files to align. The first choice is to employ the Gale–Church algorithm to align the parallel text/chunk under consideration, score the resulting 1–1 alignments using LaBSE and choose the highest-scoring alignment as a hard delimiter. If the parallel files are very large, running Gale–Church will take an excessive amount of time so SentAlign uses a fallback approach. When file size surpasses a second threshold, it resorts to a greedy algorithm that calculates LaBSE scores for 1–1 alignments in the allowed range and selects the highest one. When the hard-delimiter is found, the parallel text is split into two chunks. If the chunks are still too large, the process is repeated until all chunks of parallel text have the desired search space size.

3.4 Readjusting the Path

Thompson and Koehn (2019) argue that sentence alignment should seek a minimal parallel pair, the pair having the fewest mergers while still being truly parallel. They find that dynamic programming with cosine similarity favours many-to-many alignments over 1–1 alignments, an effect we also find when using the scoring and alignment mechanism described above. To counteract this and produce more accurate alignments, SentAlign finishes by re-evaluating each alignment in the selected path by taking another look at mergers, insertions and deletions.

First, SentAlign investigates all $n \times m$ alignments, where $(n > 1)$ and $(m > 1)$, and searches for the highest-scoring alignment which is a sub-

set of the one being investigated. If one is found that has a higher score than the original alignment, SentAlign amends the alignment path to add that as well as any other sentence pairs scoring above S_{min}. If any sentences are left they are added to the list of null alignments, containing previous insertions and deletions. Second, SentAlign looks at the list of non-aligned source and target sentences, i.e. null alignments. If a non-aligned sentence is adjacent to a sentence which has been aligned, SentAlign tries merging it to that alignment and calculates the LaBSE score. If the score increases, the path is amended. This is repeated until no more amendments can be made.

When the re-evaluation is finished, SentAlign writes out the set of alignments generated by the selected path through the alignment graph.

4 Evaluation

We evaluated SentAlign by comparing the system to other sentence aligners, both using sentence alignment evaluation sets and by testing the impact on downstream MT task.

4.1 Two evaluation sets

We compared SentAlign to five other sentence aligners: Vecalign, Bleualign, Gargantua, Hunalign and Gale-Church (using their default settings). We used two evaluation sets:

1. The manually aligned German–French evaluation set created from the Text+Berg corpus (Volk et al., 2010), first used to evaluate Bleualign and commonly used for sentence alignment evaluation since then.

2. We compiled an evaluation set for English–Icelandic sentence alignment from 10 aligned documents in five subcorpora of the ParIce corpus (Barkarson and Steingrímsson, 2019). The evaluation set (Steingrímsson, 2021) is available under an open licence and contains a total of 549 sentence alignments.[2] These documents are arguably easier to align than the Text+Berg documents, as none of them contain long stretches of non-alignments and there are few n–m merging alignments.

When translating the evaluation sets for Bleualign, we use OPUS-MT[3] (Tiedemann and Thottingal, 2020).

[2] http://hdl.handle.net/20.500.12537/150
[3] https://opus.nlpl.eu/Opus-MT/

Alignment results on Text+Berg						
	Strict			Lax		
Algorithm	P	R	F_1	P	R	F_1
Gargantua	0.76	0.75	0.76	0.89	0.78	0.83
Hunalign	0.66	0.69	0.67	0.86	0.74	0.80
Gale–Ch.	0.68	0.69	0.69	0.80	0.73	0.76
Vecalign	0.90	0.90	0.90	0.99	0.91	0.95
Bleualign	0.93	0.66	0.77	**1.00**	0.68	0.81
SentAlign	**0.94**	**0.93**	**0.93**	**1.00**	**0.93**	**0.96**

Table 1: Evaluating on the German–French Text+Berg evaluation set. The highest scores are in bold. SentAlign outperforms all systems both for the strict and lax conditions, although Bleualign has a perfect score for precision, just like SentAlign.

We used the development set from the Text+Berg corpus to search for the best parameters for SentAlign. We found the best S_{min} (LaBSE) threshold to be 0.4, maximum number of words per language before applying a length penalty to be 80, and the penalty for each word exceeding that maximum to be 0.01. We performed a complete search through the alignment matrix, without chunking the search space by finding anchors as all the evaluation files were within the limits for the hard delimiters.

While none of the aligners used, with the exception of Bleualign, allow reordering of sentences in cases of possible crossing alignments, there are examples of such alignments in the Text+Berg evaluation set, which makes it impossible for other aligners to attain a perfect score. Furthermore, a few entries of null alignments are missing from the files distributed with Bleualign. To maintain consistency with previous reported scores, we did not make any changes to the evaluation set. As only some null alignments are included in the evaluation set and some are not, the results can be different based on whether a given sentence aligner returns null alignments or only useful alignments. We thus only calculated precision on non-null alignments, i.e. alignments that are true sentence pairs.

Following the original Bleualign paper, in Table 1 we report results both under the strict condition where exact matches between the gold alignment and the hypothesis are demanded, and under the lax condition where a hypothesis is true if there is an overlap with a gold alignment on both language sides. Under the lax condition, a 2–2 alignment, which is recognized as two 1–1 alignments, will yield two true positives, while it would yield two false positives under the strict condition.

We use the same settings and parameters as before for all the aligners when we evaluate on the

Alignment results on English–Icelandic evaluation set						
	Strict			Lax		
Algorithm	P	R	F_1	P	R	F_1
Gargantua	0.82	0.76	0.79	0.89	0.78	0.83
Hunalign	0.72	0.75	0.73	0.87	0.78	0.82
Gale–Ch.	0.78	0.79	0.79	0.87	0.81	0.84
Vecalign	0.92	0.94	0.93	0.97	0.95	0.96
Bleualign	0.93	0.78	0.85	0.98	0.79	0.88
SentAlign	**0.95**	**0.95**	**0.95**	**0.99**	**0.96**	**0.97**

Table 2: Evaluating on the English–Icelandic evaluation set. The highest scores are in bold. SentAlign outperforms other systems and Vecalign is the only other aligner that comes close.

English–Icelandic evaluation set. As with the evaluation set from Text+Berg, the sentence embeddings-based alignment systems SentAlign and Vecalign are the most accurate. Using this evaluation set, the scores are higher for all aligners (see Table 2). Even though we are missing a development set for the en–is language pair and used the SentAlign parameters set for the Text+Berg de–fr development set, SentAlign does well. The results might possibly improve even more if we were to search for the best values for this sort of en–is data as the acceptance threshold for LaBSE may be different for different language pairs. While we found that 0.4 was the optimum threshold score for the Text+Berg corpus, Feng et al. (2022) set their threshold when mining sentences from CommonCrawl to 0.6. This suggests that analysis of the languages to be processed could be useful on a case-by-case basis.

4.2 Downstream MT

For the downstream MT task, we aligned English and Icelandic documents containing EEA regulations and directives. These documents are available as a subcorpus of ParIce 21.10[4] which is published with an evaluation set in that domain.[5] We used fairseq (Ott et al., 2019) to train Transformer$_{\text{BASE}}$ models (Vaswani et al., 2017), and SacreBleu (Post, 2018) to calculate BLEU scores and statistical significance using the pairwise bootstrap test (Koehn, 2004). Table 3 reports the results for all systems, showing that SentAlign achieved the best results of the six aligners evaluated, with BLEU scores of 42.8 and 53.6, for en→is and is→en, respectively. A significance test shows that this is significantly better than all the other aligners.

[4] http://hdl.handle.net/20.500.12537/145
[5] http://hdl.handle.net/20.500.12537/146

Downstream MT Task			
Sentence Aligner	no. pairs	en→is	is→en
Gargantua	606,768	39.1	48.9
Hunalign	717,879	41.4	52.1
Gale–Church	683,813	41.8	51.4
Vecalign	670,595	41.8	51.7
Bleualign	627,019	42.0	53.0
SentAlign	877,485	***42.8***	***53.6***

Table 3: Results for MT systems trained on sentence pairs generated by different alignment tools. The table shows number of aligned pairs generated by the tools and BLEU scores for the MT systems. Bold and italic scores are the highest scores for each category and significantly higher than other systems.

5 Conclusion

SentAlign is an accurate, scalable and easy-to-use sentence alignment system. It uses the LaBSE model, which has been trained to generate sentence embeddings in 109 languages, to score alignment candidates. The alignment algorithm considers all possible paths through the alignment graph where the number of merges for adjoining sentences in each language is under a user-set threshold, and the maximum number of nodes in the search space is less than the DaC-threshold. Evaluation on two sentence alignment evaluation sets, as well as on a downstream MT task, show that the aligner is highly competitive, outperforming other alignment systems in most regards. SentAlign is distributed under an Apache 2.0 licence.

Limitations

SentAlign can deliver accurate results for medium to high-resource languages in common scenarios. It is capable of evaluating all possible alignment paths through the alignment graph for parallel documents. However, as the documents get larger this may be at the cost of speed and, for very large documents, alignment time would be too long for practical use. To address this, our DaC-mechanism is applied, which enables the alignment of very large documents within reasonable time limits. Nevertheless, we can expect the system to run into problems when the number of lines in each document reaches multiple tens of thousands, due to memory constraints as well as the time factor.

LaBSE is trained on 109 languages. As noted in Section 4.1, the optimal minimum score threshold may be different between language pairs, impacting insertions and deletion made by the aligner. Furthermore, we can expect the accuracy of our

scoring function to fall if the tool is used on languages not represented in the LaBSE training data.

Finally, we used the default OPUS-MT models for aligning with Bleualign. By replacing the OPUS-MT models with higher quality models, the results for Bleualign may be further improved.

Acknowledgements

This work was supported by the The Icelandic Centre for Research, RANNIS grant number 228654-051, and by the ADAPT Centre for Digital Content Technology which is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Starkaður Barkarson and Steinþór Steingrímsson. 2019. Compiling and Filtering ParIce: An English-Icelandic Parallel Corpus. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 140–145, Turku, Finland. Linköping University Electronic Press.

Fabienne Braune and Alexander Fraser. 2010. Improved Unsupervised Sentence Alignment for Symmetrical and Asymmetrical Parallel Corpora. In *Coling 2010: Posters*, pages 81–89, Beijing, China. Coling 2010 Organizing Committee.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of Bleu in Machine Translation Research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, volume 32, page 7059–7069, Vancouver, Canada. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Edsger W Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

William A. Gale and Kenneth W. Church. 1991. A Program for Aligning Sentences in Bilingual Corpora. In *29th Annual Meeting of the Association for Computational Linguistics*, pages 177–184, Berkeley, California. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the Impact of Various Types of Noise on Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich and Martin Volk. 2010. MT-based Sentence Alignment for OCR-generated Parallel Texts. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas: Research Papers*, Denver, Colorado. Association for Machine Translation in the Americas.

Rico Sennrich and Martin Volk. 2011. Iterative, MT-based Sentence Alignment of Parallel Texts. In *Proceedings of the 18th Nordic Conference of Computational Linguistics*, pages 175–182, Riga, Latvia. Northern European Association for Language Technology (NEALT).

Steinþór Steingrímsson. 2021. Icelandic-English test set for sentence alignment 21.10. CLARIN-IS.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved Sentence Alignment in Linear Time and Space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Dániel Varga, Péter Halácsy, András Kornai, Nagy Viktor, Nagy László, Németh László, and Tron Viktor. 2005. Parallel corpora for medium density languages. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, pages 590–596, Borovets, Bulgaria. INCOMA Ltd.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5999–6009, Long Beach, California.

Martin Volk, Noah Bubenhofer, Adrian Althaus, Maya Bangerter, Lenz Furrer, and Beni Ruef. 2010. Challenges in Building a Multilingual Alpine Heritage Corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 1653–1659, Valletta, Malta. European Language Resources Association (ELRA).

Wu Zhang. 2022. Improve Sentence Alignment by Divide-and-conquer. *ArXiv*, abs/2201.06907.