Vecalign: Improved Sentence Alignment in Linear Time and Space

Brian Thompson
Johns Hopkins University
brian.thompson@jhu.edu

Philipp Koehn
Johns Hopkins University
phi@jhu.edu

Abstract

We introduce Vecalign, a novel bilingual sentence alignment method which is linear in time and space with respect to the number of sentences being aligned and which requires only bilingual sentence embeddings. On a standard German–French test set, Vecalign outperforms the previous state-of-the-art method (which has quadratic time complexity and requires a machine translation system) by 5 F_1 points. It substantially outperforms the popular Hunalign toolkit at recovering Bible verse alignments in medium- to low-resource language pairs, and it improves downstream MT quality by 1.7 and 1.6 BLEU in Sinhala→English and Nepali→English, respectively, compared to the Hunalign-based Paracrawl pipeline.

1 Introduction

Sentence alignment is the task of taking parallel documents, which have been split into sentences, and finding a bipartite graph which matches minimal groups of sentences that are translations of each other (see Figure 1). Following prior work, we assume non-crossing alignments but allow local sentence reordering within an alignment.

Sentence-aligned bitext is used to train nearly all machine translation (MT) systems. Alignment errors have been noted to have a small effect on statistical MT performance (Goutte et al., 2012). However, misaligned sentences have been shown to be much more detrimental to neural MT (NMT) (Khayrallah and Koehn, 2018).

Sentence alignment was a popular research topic in the early days of statistical MT, but received less attention once standard sentence-aligned parallel corpora became available. Interest in low-resource MT has led to a resurgence in data gathering methods (Buck and Koehn, 2016; Zweigenbaum et al., 2018; Koehn et al., 2019), but we find limited recent work on bilingual sentence alignment.

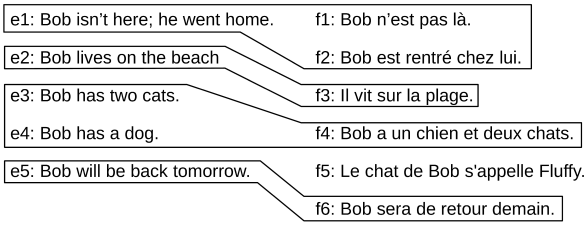

Figure 1: Sentence alignment takes sentences $e_1,...,e_N$ and $f_1,...,f_M$ and locates minimal groups of sentences which are translations of each other, in this case (e_1)-(f_1, f_2), (e_2)-(f_3), (e_3,e_4)-(f_4), and (e_5)-(f_6).

Automatic sentence alignment can be roughly decomposed into two parts:

1. A score function which takes one or more adjacent source sentences and one or more adjacent target sentences and returns a score indicating the likelihood that they are translations of each other;
2. An alignment algorithm which, using the score function above, takes in two documents and returns a hypothesis alignment.

We improve both parts, presenting (1) a novel scoring function based on normalized cosine distance between multilingual sentence embeddings, in conjunction with (2) a novel application of a dynamic programming approximation (Salvador and Chan, 2007) which makes our algorithm linear in time and space complexity with respect to the number of sentences being aligned. We release a toolkit containing our implementation.[1]

Our method outperforms previous state-of-the-art, which has quadratic complexity, indicating that our proposed score function outperforms prior work and the approximations we make in alignment are sufficiently accurate.

[1] https://github.com/thompsonb/vecalign

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1342–1348, Hong Kong, China, November 3–7, 2019.

2 Related Work

Early sentence aligners (Brown et al., 1991; Gale and Church, 1993) use scoring functions based only on the number of words or characters in each sentence and alignment algorithms based on dynamic programming (DP; Bellman, 1953). DP is O(NM) time complexity, where N and M are the number of sentences in the source and target documents. Later work added lexical features and heuristics to speed up search, such as limiting the search space to be near the diagonal (Moore, 2002; Varga et al., 2007). More recent work introduced scoring methods that use MT to get both documents into the same language (Bleualign; Sennrich and Volk, 2010) or use pruned phrase tables from a statistical MT system (Coverage-Based; Gomes and Lopes, 2016). Both methods "anchor" high-probability 1–1 alignments in the search space and then fill in and refine alignments. Locating anchors is $O(NM)$ time complexity.

3 Method

We propose a novel sentence alignment scoring function based on the similarity of bilingual sentence embeddings. A distinct but non-obvious advantage of sentence embeddings is that blocks of sentences can be represented as the average of their sentence embeddings. The size of the resulting vector is not dependent on the number of sentence embeddings being averaged, thus the time/space cost of comparing the similarity of blocks of sentences *does not depend on the number of sentences being compared*. We show empirically (see § 4.2) that average embeddings for blocks of sentences are sufficient to produce approximate alignments, even in low-resource languages. This enables us to approximate DP in O($N+M$) in time and space.

3.1 Bilingual Sentence Embeddings

We propose to use the similarity between sentence embeddings as the scoring function for sentence alignment. Sentence embedding similarity has been shown effective at filtering out non-parallel sentences (Hassan et al., 2018; Chaudhary et al., 2019) and locating parallel sentences in comparable corpora (Guo et al., 2018). We use the publicly available LASER multilingual sentence embedding method (Artetxe and Schwenk, 2018) and model, which is pretrained on 93 languages. However, our method is not specific to LASER.

3.2 Scoring Function

Cosine similarity is an obvious choice for comparing embeddings but has been noted to be globally inconsistent due to "hubness" (Radovanović et al., 2010; Lazaridou et al., 2015). Guo et al. (2018) proposed a supervised training approach for calibration, and Artetxe and Schwenk (2019) proposed normalization using nearest neighbors. We propose normalizing instead with *randomly* selected embeddings as it has linear complexity. Sentence alignment seeks *minimal* parallel units, but we find that DP with cosine similarity favors many-to-many alignments (e.g. reporting a 3–3 alignment when it should report three 1–1 alignments). To remedy this issue, we scale the cost by the number of source and target sentences being considered in a given alignment. Our resulting scoring cost function is:

$$c(x,y) = \frac{(1-\cos(x,y))\,\mathrm{nSents}(x)\,\mathrm{nSents}(y)}{\sum_{s=1}^{S} 1-\cos(x,y_s) + \sum_{s=1}^{S} 1-\cos(x_s,y)}$$

where x, y denote one or more sequential sentences from the source/target document; $\cos(x,y)$ is the cosine similarity between embeddings[2] of x, y; $\mathrm{nSents}(x)$, $\mathrm{nSents}(y)$ denote the number of sentences in x, y; and $x_1,...,x_S$, $y_1,...,y_S$ are sampled uniformly from the given document.

Following standard practice, we model insertions and deletions in DP using a skip cost c_{skip}. The raw value of c_{skip} is only meaningful when compared to other costs, thus we do not expect it to generalize across different languages, normalizations, or resolutions. We propose specifying instead a parameter β_{skip} which defines the skip cost in terms of the distribution of 1–1 alignment costs at alignment time: $c_{skip} = \mathrm{CDF}^{-1}(\beta_{skip})$. CDF is an estimate of the cumulative distribution function of 1–1 alignments obtained by computing costs of randomly selected source/target sentences pairs.

3.3 Recursive DP Approximation

Instead of searching all possible sentence alignments via DP, consider first averaging adjacent pairs of sentence embeddings in both the source and target documents, halving the number of embeddings for each document. Aligning these vectors via DP (each of which are averages of 2 sentence embeddings) produces an *approximate* sen-

[2] If multiple sentences are considered on one side, they are concatenated together before embedding.

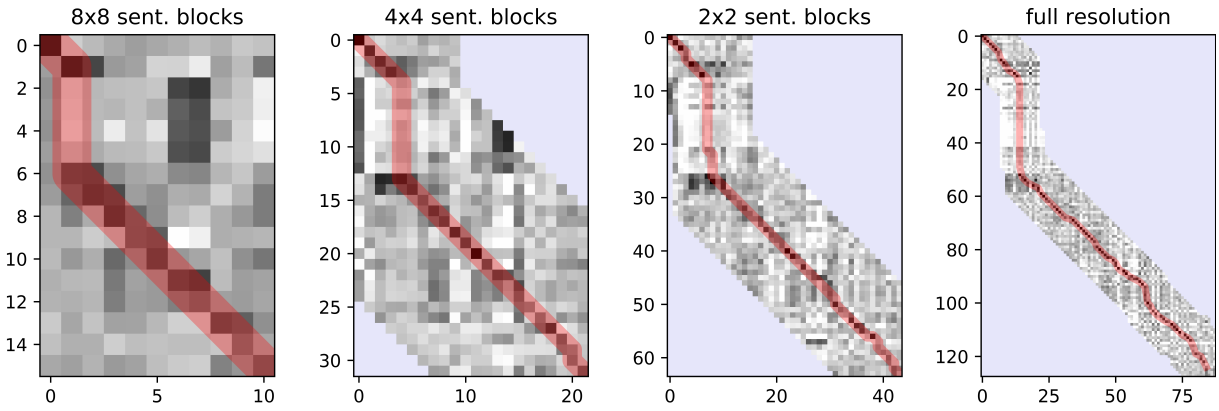

Figure 2: 1–1 alignment costs (darker = lower) for the first 88 De lines (x-axis) and 128 Fr lines (y-axis) at 4 different resolutions. The red highlight denotes alignment found by DP. The algorithm only searches near the path found at previous resolutions; light blue regions are excluded. The vertical part of the path in the top left of each plot is due to 36 extra lines being present in the Fr document. Window size is increased for visualization purposes.

tence alignment, at a cost of $\left(\frac{N}{2}\right)\left(\frac{M}{2}\right)$ comparisons. We can then refine this approximate alignment using the original sentence vectors, constraining ourselves to a small window around the approximate alignment. At a minimum, we must search a window size w large enough to consider all paths covered by the lower-resolution alignment path, but w can also be increased to allow recovery from small errors in the approximate alignment.[3] The length of the refinement path to search is at most $N+M$ (all deletions/insertions), so refining the path requires at most $(N+M)w$ comparisons. Thus the full NM comparisons can be approximated by $(N+M)w+\left(\frac{N}{2}\right)\left(\frac{M}{2}\right)$ comparisons. Applied recursively,[4] we can approximate our quadratic NM cost with a sum of linear costs:

$$(N+M)w+\left(\frac{N}{2}+\frac{M}{2}\right)w+\left(\frac{N}{4}+\frac{M}{4}\right)w+\ldots$$
$$=\sum_{k=0,1,2,\ldots}\frac{(N+M)w}{2^k}=2(N+M)w$$

See Figure 2 for an illustration of this method. We consider only insertions, deletions, and 1–1 alignments in all but the final search. Recursive down sampling and refining of DP was proposed for dynamic time warping in Salvador and Chan (2007), but has not previously been applied to sentence alignment. We direct the reader to that work for a more formal analysis showing the time/space complexity is linear.

[3] We use $w=10$ for all experiments in this work.

[4] In practice, we compute the full DP alignment once the down sampled sizes are below an acceptably small constant. We also find vectors for large blocks of sentences become correlated with each other, so we center them around $\vec{0}$.

4 Experiments & Results

4.1 Text+Berg Alignment Accuracy

We evaluate sentence alignment accuracy using the development/test split released with Bleualign, consisting of manually aligned yearbook articles published in both German and French by the Swiss Alpine Club from the Text+Berg corpus (Volk et al., 2010). Hyperparameters were chosen to optimize F_1 on the development set. We consider alignments of up to 6 total sentences; that is we allow alignments of size Q–R where $Q+R\leq 6$.

We compare to Gale and Church (1993), Moore (2002), Hunalign (Varga et al., 2007), Bleualign (Sennrich and Volk, 2010), Gargantua (Braune and Fraser, 2010), and Coverage-Based (Gomes and Lopes, 2016). We run Hunalign in both bootstrapping mode as well as using a publically available De–Fr lexicon from OPUS (Tiedemann, 2012)[5] created from Europarl (Koehn, 2005). Since Bleualign depends on the quality of MT output, we re-run it with a modern NMT system.[6]

Our proposed method outperforms the next best method (Coverage-Based) by 5 F_1 points: see Table 1. Gargantua and bootstrapped Hunalign have both been reported to perform well (Abdul-Rauf et al., 2012); this dataset may be too small to bootstrap good lexical features.[7] Bleualign improves by 3 F_1 points by using an NMT system.

[5] https://object.pouta.csc.fi/OPUS-Europarl/v7/dic/de-fr.dic.gz

[6] https://docs.microsoft.com/en-us/azure/cognitive-services/translator/

[7] We run only on the test/development articles, not the full Text+Berg corpus.

Algorithm	O( )	P	R	F_1
Gargantua	N^2	0.48	0.54	0.51
Hunalign w/o lexicon	$\boldsymbol{N}$	0.59	0.70	0.64
Hunalign w/ lexicon	$\boldsymbol{N}$	0.61	0.73	0.66
Gale and Church (1993)†	N^2	0.71	0.72	0.72
Moore (2002)†	‡	0.86	0.71	0.78
Bleualign†	N^2	0.83	0.78	0.81
Bleualign-NMT	N^2	0.85	0.83	0.84
Coverage-Based*	N^2	0.85	0.84	0.85
Vecalign	$\boldsymbol{N}$	**0.89**	**0.90**	**0.90**

Table 1: De–Fr test precision (P), recall (R), and F_1. *best reported in Gomes and Lopes (2016). †Best reported in Sennrich and Volk (2010). ‡O( ) is data dependent. We assume $N = M$ for simplicity.

Language	ISO 639-1	Bible # Sents	LASER # Train Lines
Arabic	Ar	45980	8.2M
Turkish	Tr	48492	5.7M
Somali	So	37413	85k
Afrikaans	Af	37081	67k
Tagalog	Tl	34207	36k
Norwegian	No	37064	0*

Table 2: Bible statistics. *LASER was not trained on Norwegian but appears to generalize to it.

4.2 Bible Alignment Accuracy

We are unaware of a multilingual, low resource, parallel dataset with human sentence-level annotations. As a substitute for gold standard sentence alignment, we use Bible verse alignment and sentence split each verse.[8]

The Bible has a number of properties which make it appealing for sentence alignment evaluation: It is much larger than existing sentence alignment test sets, and it is multi-way parallel in a large number of languages. Bibles are not aligned at the sentence level, but contain verse marking denoting segments typically on the scale of a partial sentence to a few sentences. This creates two potential issues for sentence alignment evaluation: First, a single sentence may span more than one verse. Inspecting the English Bible suggests that this is rare, and sentence aligners should be able to handle occasional over-segmentation of sentences as in practice they are run on errorful automatic sentence segmentation. Second, a verse may contain more than one sentence. This is problematic when it happens on both languages being aligned, since the true sentence alignment cannot be determined (e.g., a verse which is two sentences in each language could be two 1–1 alignments or one 2–2 alignment). To evaluate with verse-level annotations, we propose converting the sentence alignment output into verse alignments by combining any consecutive sentence alignments for which all sentences in the alignments, on both the source and target side, came from the same verse. We report F_1 compared to the gold-standard verse alignments, denoting it as *verse-level* F_1 to distinguish it from F_1 computed at the sentence level.

Languages	Verse-level F_1 Vecalign	Hunalign
Af–Ar	**0.863**	0.339
Af–Tl	**0.922**	0.775
Ar–No	**0.787**	0.406
Ar–So	**0.634**	0.067
Tr–So	**0.533**	0.331
No–So	**0.697**	0.687
So–Af	**0.782**	0.738
Tl–No	**0.874**	0.764
Tr–Af	**0.703**	0.401
Tr–Tl	**0.647**	0.247

Table 3: Bible verse alignment results.

We select six languages for which Christodouloupoulos and Steedman (2015) contains a full Bible: see Table 2. Languages were chosen to provide a range of amounts of training data used in LASER.[9] From those six languages, we randomly select 10 language pairs for testing. All parameters are kept the same as § 4.1 except we only consider alignments of up to 4 total sentences. We compare to Hunalign, run in bootstrap mode, as it is the only toolkit we tried which was robust enough to run on documents of this size. Results are shown in Table 3.

On average, we see an improvement of 28 verse-level F_1 points over Hunalign. In manual analysis of the alignments we find large stretches where the Hunalign alignments are nowhere near the gold alignment in the language pairs with verse-level $F_1 < 0.35$. By contrast, errors in the proposed method are predominately local, indicating success of Vecalign's recursive DP approximation even for very long documents in low-resource languages.

[8]There is no clear choice for sentence segmentation in low-resource languages. We use https://github.com/berkmancenter/mediacloud-sentence-splitter, falling back on English for unsupported languages.

[9]Data amounts are all between the given language and English. LASER used no bitext in the language pairs under test.

4.3 Improvements to Downstream MT

One of the primary applications of sentence alignment is creating bitext for training MT systems. To test Vecalign's impact on downstream MT quality, we re-align noisy, web-crawled data in two low-resource language pairs: Sinhala–English and Nepali–English. The data is collected via Paracrawl[10] and is very similar to that released in the WMT 2019 sentence filtering task (Koehn et al., 2019), but some new data has been collected and a small amount of data was lost due to a hard disk failure. Our baseline is the standard Paracrawl pipeline using Hunalign in conjunction with a dictionary extracted from the clean data released in the shared task.

We filter the output of Vecalign and Hunalign following (Chaudhary et al., 2019), including filtering out sentences with the wrong languages and sentences with high token overlap, as this was the best performing method from the shared task.[11] We train and evaluate NMT models following the procedure/hyperparameters from the shared task.

Results are shown in Figure 3. Using Vecalign, we see improvements of 1.7 and 1.6 BLEU for the best data sizes in Sinhala→English and Nepali→English, respectively, compared to the systems trained on Hunalign output.

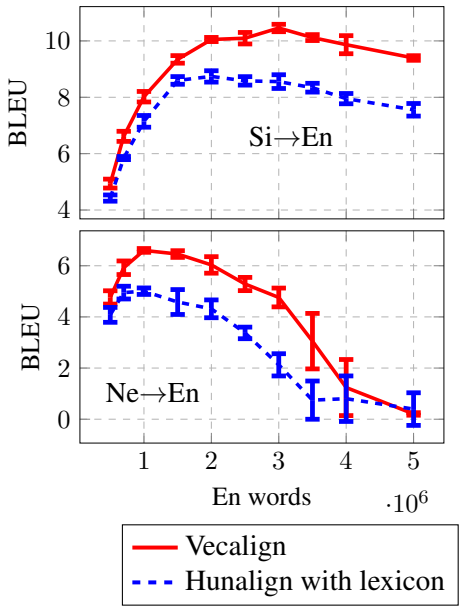

Figure 3: sacreBLEU scores (mean +/- standard deviation for 5 training runs) on FLoRes test sets for systems trained on data aligned with Vecalign vs Hunalign.

[10] https://paracrawl.eu/

[11] We use the publicly available multilingual LASER model, which is not trained on Nepali.

5 Empirical Runtime Analysis

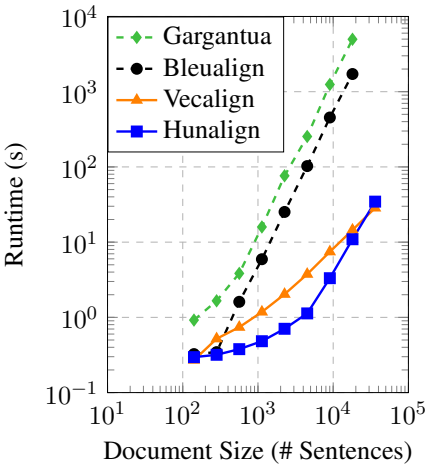

Figure 4: Time required to align various portions of En→Hu Bibles, for various systems. Plot is logarithmic in both runtime and number of sentences, thus a slope of one (i.e. runtime doubles each time the number of sentences doubles) indicates $O(N)$, while a slope of two (i.e. runtime quadruples each time the number of sentences doubles) indicates $O(N^2)$.

Time required to align documents of various sizes are shown for Vecalign, Bleualign, Gartantua, and Hunalign in see Figure 4. As expected, Vecalign has approximately linear runtime characteristics. We use truncated portions of Hu–En Bibles in order to use the dictionary provided with Hunalign. Bleualign is run on NMT output. Vecalign settings match § 4.2. Experiments are run on a Thinkpad T480 with 32GB RAM. Times do not include translation (Bleualign), lexicon building (Hunalign), or sentence embedding (Vecalign). For reference, producing embeddings for 32k sentences, including overlaps, in each language took ~120 s on a GeForce RTX 2080 Ti GPU. Bleualign and Gargantua run out of memory on 32k sentences. Hunalign and Vecalign use ~1GB and are both very fast, aligning 32k sentences in ~30 s.

6 Conclusions

We present Vecalign, a novel sentence alignment method based on similarity of sentence embeddings and a DP approximation which is fast even for long documents. Our method has state-of-the-art accuracy in high and low resource settings and improves downstream MT quality.

Acknowledgments

Brian Thompson is supported through the National Defense Science and Engineering Graduate (NDSEG) Fellowship Program.

References

Sadaf Abdul-Rauf, Mark Fishel, Patrik Lambert, Sandra Noubours, and Rico Sennrich. 2012. Extrinsic evaluation of sentence alignment systems. *Workshop on Creating Cross-language Resources for Disconnected Languages and Styles*, pages 6–10.

Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *arXiv preprint arXiv:1812.10464*.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Richard Bellman. 1953. An introduction to the theory of dynamic programming. Technical report, RAND Corporation, Santa Monica, CA.

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, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, ACL '91, pages 169–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christian Buck and Philipp Koehn. 2016. Findings of the WMT 2016 bilingual document alignment shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 554–563, Berlin, Germany. Association for Computational Linguistics.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmn, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 263–268, Florence, Italy. Association for Computational Linguistics.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the Bible in 100 languages. *Language resources and evaluation*, 49(2):375–395.

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

Luís Gomes and Gabriel Pereira Lopes. 2016. First steps towards coverage-based sentence alignment. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2228–2231, Portorož, Slovenia. European Language Resources Association (ELRA).

Cyril Goutte, Marine Carpuat, and George Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *The Tenth Biennial Conference of the Association for Machine Translation in the Americas (AMTA 2012)*.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 165–176, Belgium, Brussels. Association for Computational Linguistics.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic Chinese to English news translation. *arXiv preprint arXiv:1803.05567*.

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. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation summit*, volume 5, pages 79–86.

Philipp Koehn, Francisco Guzmn, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 56–74, Florence, Italy. Association for Computational Linguistics.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 270–280, Beijing, China. Association for Computational Linguistics.

Robert C Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep):2487–2531.

Stan Salvador and Philip Chan. 2007. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580.

Rico Sennrich and Martin Volk. 2010. MT-based sentence alignment for OCR-generated parallel texts. In *The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292:247.

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 conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Languages Resources Association (ELRA).

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2018. Overview of the third BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of 11th Workshop on Building and Using Comparable Corpora*, pages 39–42.