Optimal Transport for Document Alignment based on Overlapping Fixed-Length Segments

Xiaotian Wang[1] Takehito Utsuro[1] Masaaki Nagata[2]
[1]Deg. Prog. Sys.&Inf. Eng., Grad. Sch. Sci.&Tech., University of Tsukuba
[2]NTT Communication Science Laboratories, NTT Corporation, Japan
[1]s2320811_@_u.tsukuba.ac.jp [1]utsuro_@_iit.tsukuba.ac.jp
[2]masaaki.nagata_@_ntt.com

Abstract

To acquire large-scale parallel corpora for NLP tasks such as Neural Machine Translation, web crawling has emerged as a popular methodology. When aligning with documents in various languages obtained through web crawling, the Sentence Movers' Distance (SMD)[1, 2] based on Optimal Transport (OT) has shown promising performance. However, we observed that compared to the SMD method using original web-crawled sentences, SMD based on overlapping fixed-length segments results in a significant improvement. Simultaneously, we conducted accuracy and speed comparisons of this approach with other conventional methods, and proposed a novel approach utilizing multiple feature vectors to represent documents.

1 Introduction

For tasks such as Neural Machine Translation, a substantial amount of parallel corpora is required during the training phase. Web crawling is an efficient approach to allow researchers to gather a large-scale parallel dataset, such as ParaCrawl Dataset[3] and JParaCrawl Dataset[4].

When dealing with a collection of documents in different languages obtained through web crawling, the initial step is document alignment, which can be broadly categorized into three strategies, URL matching[5, 6, 7], methods based on machine translation[7, 8, 9, 10], and leveraging sentence embeddings[1, 2, 5, 11]. The core concept of the last one involves transforming the sentences within documents into a series of feature vectors. These vectors are then employed to calculate the similarity between documents from different languages, with pairs exhibiting high similarity selected as alignment results. However, it should be noted that crawled documents may not have uniform sentence segmentation, as they often depend on the textual presentation format of the website. In this case, we explore an alternative approach for subdivision, which involves concatenating all the sentences included in the document at first, and then utilizing a fixed-length sliding window to partition segments, with a specified proportion of overlap between adjacent segments.

In summary, our contributions are as follows:

- Instead of using the original web-crawled sentences, we demonstrated the impact of employing overlapping fixed-length segments to generate the sequence of sentence-level embeddings for the documents.
- We proposed and showed the effect of a method for calculating document similarity through the two feature vectors.

2 Related Work

Among the various web crawling methods, Bitextor[1)][8] is one of the most widely adopted tools. Additionally, it incorporates a module known as docalign, which employs a TF-IDF strategy to score document pairs within one language through machine translation of documents in other languages.

LaBSE model[12], a pre-trained sentence embedding model, has the capability to map sentences from different languages into a unified vector space, exhibiting state-of-the-art performance in sentence embedding task and downstream applications.

The application of Optimal Transport (OT) in cross-lingual alignment, initially performing sentence-level alignment based on word embeddings, known as Word

1) https://github.com/bitextor/bitextor

Movers' Distance (WMD)[13]. With the emergency of multi-lingual sentence embedding models[12, 14, 15, 16], analogous to the WMD, Sentence Movers' Distance[1, 2] based on Optimal Transport (OT) was introduced for document-level alignment.

3 Document Alignment

3.1 Machine Translation based Document Alignment

In this paper, we utilize the docalign module[2] of Bitextor as a baseline for document alignment. It tokenizes the target language documents to create a vocabulary, and then for each word within each target language document, calculates its TF-IDF value, resulting in a feature vector of the same length as the vocabulary. Next, machine translated document is used as a query, and after tokenization, the TF-IDF values of each word within the vocabulary are computed, resulting in another feature vector for the query. The cosine similarity between these feature vectors is then computed as a measure of document similarity.

When calculating the time consumption for using docalign of Bitextor, it includes data preprocessing as well as the time required for machine translation. In this paper, we employed the pre-trained JParaCrawl-v3.0-big model[3] [4] based on fairseq toolkit[17] for machine translation. During data preprocessing, all sentences are concatenated from source language documents for simultaneous processing, and after translation, we split the results based on the original documents' order and number of sentences.

3.2 Sentence Embedding based Document Alignment

Overlapping Fixed-Length Segmentation For any given document, instead of using original web-crawled sentences to generate embeddings, we create segments by concatenating all sentences within the document into a text, then tokenizing it by the tokenizer of the LaBSE model, subsequently dividing it into segments through a fixed-length sliding window, with a specified proportion of overlap between adjacent segments.

Language-Pair Dependent Overlapping Fixed-Length Segmentation While applying the above-mentioned segmentation method, we use the same fixed-length for segmenting documents in both the source language and the target language. However, it is commonly observed that different languages may require different numbers of tokens to convey the same meaning. For instance, the English sentence "I like dogs" requires only 3 tokens, while the Japanese sentence "私 は 犬 が 好き だ" ("I like dog") needs 6 tokens. Therefore, it is worth considering whether using different fixed-lengths would result in a more natural segmentation. With this perspective, we propose a language-pair dependent proportion ρ to split the target language document with fixed-length ρL when segmenting the source language document using a fixed-length L.

For any document A in the source language and any document B in the target language, we employ the LaBSE model[12] to perform length-768 dense sentence-level embedding, resulting in two sets of vectors, $\{e_{A,i}\}$ and $\{e_{B,j}\}$, while $e_{*,i} \in \mathbb{R}^{768}$, representing the ith segment's embedding in document $*$. We employ the following three methods to calculate document pair similarity and compare our proposed segmentation strategy with the use of the original sentences crawled from websites.

3.2.1 Mean-Pooled Vector based Method

The most straightforward approach is to use the mean-pooled vectors from the sets $\{e_{A,i}\}$ and $\{e_{B,j}\}$ as the feature vectors for document A and B, calculating their cosine similarity to score the document pair.

$$e_{A,mean} = \sum_{i=1}^{n} e_{A,i}/n \tag{1}$$

$$e_{B,mean} = \sum_{i=1}^{m} e_{B,i}/m \tag{2}$$

$$Docsim(A,B) = Cos(e_{A,mean}, e_{B,mean}) \tag{3}$$

where $e_{*,mean}$ represents the mean-pooled vector of document $*$, n, m represents the number of vectors in $\{e_{A,i}\}$ and $\{e_{B,j}\}$, and $Docsim(A,B)$ represents the document similarity score.

3.2.2 Multiple Feature Vectors based Method

Differing from the approach discussed in Section 3.2.1, where only the mean-pooled vector is used to represent the document, we propose a method that utilizes multiple feature vectors for calculating document similarity. The

2) https://github.com/bitextor/bitextor/tree/master/document-aligner
3) https://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/

Table 1 Information of Dataset

Domain	Marubeni	Nishi-Shinjuku	Rakuten	NTT CS	All
Num of Japanese Documents	73	16	75	68	232
Num of Aligned English Documents	73	16	75	68	232
Num of Candidate English Documents	251	42	319	319	931
Avg tokens of Japanese Documents	2447.36	340.56	3541.55	726.37	2151.35
Avg tokens of Aligned English Documents	1598.51	217.69	2174.49	441.88	1350.47

selection of these feature vectors can be achieved through various schemes, such as the first vector, the mean-pooled vector, the max-pooled vector, and the last vector of $\{e_{*,i}\}$. We aggregate these vectors to form a feature vector set $\{e_{*,f_i}\}$ for the document $*$, and the document similarity score is calculated as follows:

$$Docsim(A,B) = \sum_{k=1}^{r} \lambda_k Cos(e_{A,f_k}, e_{B,f_k}) \tag{4}$$

where r represents the number of selected features, and λ represents the weights for adjusting the reliance on features.

In this paper, we focus solely on two features from $\{e_{*,i}\}$: the first vector $e_{*,1}$ and the mean-pooled vector $e_{*,mean}$, while the calculation formula can be rewritten as follows:

$$\begin{aligned} Docsim(A,B) = {} & \lambda Cos(e_{A,1}, e_{B,1}) \\ & + (1-\lambda) Cos(e_{A,mean}, e_{B,mean}) \end{aligned} \tag{5}$$

3.2.3 Optimal Transport based Method

Optimal Transport, which is also known as Earth Movers' Distance (EMD)[18] and Wasserstein Metric, is a measure of the distance between two probability distributions. For the application in document alignment, known as Sentence Movers' Distance (SMD), it calculates the minimum cost of transforming the distribution of document A to the distribution of document B. It represents each document as a normalized *bag-of-sentences* (nBOS) where each segment has associated with its some probability mass.

Specifically, all segments from document A and document B are utilized to establish a vocabulary of size V, with the sequence of embeddings $\{v_i\}$ for the ith segment. $d_{A,i}$ is defined as the wight of ith segment of vocabulary in document A. While El-Kishky and Guzmán[2] has discussed various calculation methods for $d_{A,i}$, we adopt the assumption that gives weight to segments by relative frequencies [4], which is calculated as follows:

$$d_{A,i} = cnt(i)/|A| \tag{6}$$

where $cnt(i)$ is frequency of ith segment in document A, and $|A|$ is the total number of segments in document A.

We denote $\Delta(i,j)$ as the distance between the ith segment and jth segment in the vocabulary. Differing from Kusner et al.[13], who employed the Euclidean distance to calculate $\Delta(i,j)$, we utilize the cosine distance as a replacement. The SMD between document A and B can be calculated as follows:

$$\Delta(i,j) = 1 - Cos(i,j) \tag{7a}$$

$$SMD(A,B) = \min_{T \geq 0} \sum_{i=1}^{V} \sum_{j=1}^{V} T_{ij} \Delta(i,j) \tag{7b}$$

subject to:

$$\forall i \sum_{j=1}^{V} T_{ij} = d_{A,i} \tag{8a}$$

$$\forall j \sum_{i=1}^{V} T_{ij} = d_{B,j} \tag{8b}$$

and $T \in \mathbb{R}^{V \times V}$ is a nonnegative matrix, where each T_{ij} denotes how much of segment i in document A is assigned to segments j in document B, and constraints ensure the flow of a given segment cannot exceed its allocated mass.

4 Experiment

4.1 Dataset

We used manually aligned document pairs obtained from four websites: Marubeni, Nishi-Shinjuku, Rakuten, and NTT Computer Science. For each website, we randomly sampled a set of Japanese documents, and then made a pool of candidates for corresponding English documents on the same website using four different document alignment methods. We then manually selected the correctly corresponding English document for a Japanese document in the pool. The detailed dataset development procedure is provided in the Appendix A. As shown in Table 1, the total number of Japanese documents is 232, and the aligned English documents also amount to 232, which are included

4) We mainly refer to the program of OTalign[19] for OT calculation, which utilizes the POT[5] Python library.

5) https://pythonot.github.io/

Table 2 The final result of ja-en document alignment (F1 Score / Average time for all process (sec.)), where machine translation for docalign is from Japanese to English, the calculation object for OT is top 20 similar English documents by "LaBSE + Mean-Pool", time consumption for "MT + docalign" combines data preprocessing, machine translation and docalign, time consumption for sentence embedding based methods is composed of sentence embedding and similarity calculation, and "Fixed-Length segmentation" represents segmenting without overlapping. We put the detailed result with hyper-parameter settings in Appendix B.

Segment Strategy	F1 Score / Average time consumption for all process (sec.)			
	Web-Crawled Sentences	Fixed-Length Segmentation	Overlapping Fixed-Length Segmentation	Language-Pair Dependent Overlapping Fixed-Length Segmentation
MT + Docalign	0.7880 / **161.95s**	-	-	-
LaBSE + Mean-Pool	0.8276 / 277.65s	0.8147 / **71.72s**	0.8621 / **124.29s**	0.8707 / **124.03s**
LaBSE + 2 Features	**0.8577** / 330.86s	**0.8577** / 120.88s	**0.9009** / 177.35s	0.9009 / 176.16s
LaBSE (Faiss) + OT	0.8362 / 302.53s	0.8491 / 84.37s	0.8879 / 135.51s	**0.9224** / 137.81s

within the entire set of 931 candidate English documents. The average tokens of Japanese documents and aligned English documents are also given, aiming to help us judge the appropriate language-pair dependent proportion ρ as mentioned in Section 3.2.

4.2 Experiment Setting

We utilize LaBSE tokenizer and model[6)] [12] for tokenizing and sentence embedding. The retrieval for document pairs is from 232 Japanese documents to 931 candidate English documents. For each Japanese document, Faiss[20] search is used to find top k similar English documents based on the mean-pooled method mentioned in Section 3.2.1 as the object of calculation for OT, while for other approaches the document similarity is calculated with all the 931 candidate English documents. The final result enforces the 1-1 rule: each document should be aligned only once, and we evaluate the final result by F1 Score. All the experiments are conducted on one NVIDIA RTX A6000 GPU.

4.3 Result

As the result shown in Table 2, all sentence embedding based methods achieved F1 scores surpassing MT based docalign. Furthermore, when employing segmentation methods other than web-crawled sentences, "LaBSE + Mean-Pool" and "LaBSE (Faiss) + OT" demonstrated faster computational speeds compared to "MT + Docalign".

Among all the sentence embedding based methods, "LaBSE + Mean-Pool" exhibited the fastest speed while also displaying the lowest accuracy.

When using fixed-length segmentation without overlapping, the comparison with web-crawled sentences did not yield significant changes. However, on the contrary, when overlapping was introduced, the F1 Score was obviously improved.

The "Language-Pair Dependent Overlapping Fixed-Length Segmentation" strategy did not significantly improve "LaBSE + Mean-Pool" and "LaBSE + 2 Features", as the final document similarity was not directly correlated with any specific segment. However, it had a substantial positive impact on "LaBSE (Faiss) + OT", which utilized each segment in the computation of document similarity.

5 Analysis of Overlapping Rate

Table 3 Analysis for overlapping rate using "LaBSE (Faiss) + OT" with fixed-length $L = 100$.

Rate	F1 Score	Time (sec.) (Embedding)	Time (sec.) (Similarity)
0.0	0.8491	69.30s	15.07s
0.3	0.8836	89.85s	16.83s
0.5	0.8879	119.44s	16.07s
0.8	0.8664	276.63s	15.39s

According to the results in Table 3, There are apparent discrepancies regarding the utilization of overlapping, and the F1 Score achieves a maximum at the rate of 0.5. Furthermore, with the escalation of the overlapping rate, there is a corresponding augmentation in the temporal demand for the embedding process. Nonetheless, it appears that this exerts no substantial impact on the computation speed of optimal transport.

6 Conclusion

This paper presents a strategy for splitting documents into overlapping fixed-length segments to calculate document similarity, and proposes a method based on multiple feature vectors, which exhibits superior accuracy when contrasted with a sole reliance on the mean-pooled vector.

6) https://huggingface.co/setu4993/LaBSE

Acknowledgment

We would like to show our gratitude to Bui Tuan Thanh, the intern student at NTT Communication Science Laboratories in 2022, for assisting us in creating the dataset.

References

[1] E. Clark, A. Celikyilmaz, and N. Smith. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In **Proc 57th ACL**, pp. 2748–2760, 2019.

[2] A. El-Kishky and F Guzmán. Massively multilingual document alignment with cross-lingual sentence-mover's distance. In **Proc 1st AACL and 10th IJCNLP**, pp. 616–625, 2020.

[3] M. Bañón, P. Chen, B. Haddow, K. Heafield, H. Hoang, M. Esplà-Gomis, M. Forcada, A. Kamran, F. Kirefu, P. Koehn, S. Ortiz Rojas, L. Pla Sempere, G. Ramírez-Sánchez, E. Sarrías, M. Strelec, B. Thompson, W. Waites, D. Wiggins, and J. Zaragoza. ParaCrawl: Web-scale acquisition of parallel corpora. In **Proc 58th ACL**, pp. 4555–4567, 2020.

[4] M. Morishita, K. Chousa, J. Suzuki, and M. Nagata. JParaCrawl v3.0: A large-scale English-Japanese parallel corpus. In **Proc 13th LREC**, pp. 6704–6710, 2022.

[5] A. El-Kishky, V. Chaudhary, F. Guzmán, and P. Koehn. CCAligned: A massive collection of cross-lingual web-document pairs. In **Proc 2020 EMNLP**, pp. 5960–5969, 2020.

[6] U. Germann. Bilingual document alignment with latent semantic indexing. In **Proc 1st WMT**, pp. 692–696, 2016.

[7] L. Gomes and G. Pereira Lopes. First steps towards coverage-based document alignment. In **Proc 1st WMT**, pp. 697–702, Berlin, Germany, 2016.

[8] M. Esplà-Gomis. Bitextor: a free/open-source software to harvest translation memories from multilingual websites. In **Proc 12th MTSummit**, 2009.

[9] A. Dara and Y. Lin. YODA system for WMT16 shared task: Bilingual document alignment. In **Proc 1st WMT**, pp. 679–684, 2016.

[10] V. Shchukin, D. Khristich, and I. Galinskaya. Word clustering approach to bilingual document alignment (WMT 2016 shared task). In **Proc 1st WMT**, pp. 740–744, 2016.

[11] S. Steingrimsson. A sentence alignment approach to document alignment and multi-faceted filtering for curating parallel sentence pairs from web-crawled data. In **Proc 8th WMT**, pp. 366–374, 2023.

[12] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. Language-agnostic BERT sentence embedding. In **Proc 60th ACL**, pp. 878–891, 2022.

[13] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In **Proc 32nd PRML**, pp. 957–966, 2015.

[14] M. Artetxe and H. Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. **Journal 9th TACL**, pp. 597–610, 2019.

[15] N. Reimers and I. Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In **Proc 2020 EMNLP**, pp. 4512–4525, 2020.

[16] K. Heffernan, O. Çelebi, and H. Schwenk. Bitext mining using distilled sentence representations for low-resource languages. In **Findings of 2022 EMNLP**, pp. 2101–2112, 2022.

[17] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. fairseq: A fast, extensible toolkit for sequence modeling. In **Proc NAACL 2019**, pp. 48–53, 2019.

[18] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover' s distance as a metric for image retrieval. In **Journal 2000 IJCV**, pp. 99–121, 2000.

[19] Y. Arase, H. Bao, and S. Yokoi. Unbalanced optimal transport for unbalanced word alignment. In **Proc 61st ACL**, pp. 3966–3986, 2023.

[20] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. **Journal 2019 IEEE**, pp. 535–547, 2019.

A Dataset Development Procedure

As for the dataset development procedure, it was initially crawled from four domain websites: Marubeni, Nishi-Shinjuku, Rakuten, and NTT Computer Science, in both Japanese and English. For each website, we randomly sampled a set of Japanese documents, and then made a pool of candidates for corresponding English documents on the same website using four different document alignment methods.

- Machine Translation + BM25
- Machine Translation + TF-IDF
- URL matching
- CCAligned[5]

These candidate pairs were then manually evaluated, and the correct 1-1 document pairs were identified.

B Hyper-parameters in Experiments

We give a final result incorporating hyper-parameter settings in Table B. The language-pair dependent proportion ρ is determined by the average tokens of Japanese documents and aligned English documents as mentioned in Section 4.1. However, because we used only one test dataset in this experiment, all hyper-parameter tuning was performed on the test dataset, which is generally a process conducted on validation data.

C JparaCrawl-v3.0-big Setting

The hyper-parameters for the generator of the JparaCrawl-v3.0-big ja-en model is provided in Table 5.

Table 5 Hyper-parameters for the generator of the JparaCrawl-v3.0-big model.

Rate	F1 Score
Model	JparaCrawl-v3.0-big ja-en
Max-tokens	40,960
Beam Size	6
Lenpen	1.0
Log-format	simple
Task	translation
Remove-bpe	

Table 4 The final results of ja-en document alignment incorporating hyper-parameter settings, where "MT + docalign" represents for using web-crawled sentences, $\lambda = 0.4$ is set for "LaBSE + 2 Features" to combine $Cos(e_{A,1}, e_{B,1})$ and $Cos(e_{A,mean}, e_{B,mean})$ as mentioned in Section 3.2.2, the calculation object found by Faiss for OT is 20 most similar English documents, "Fixed-Length" represents for using fixed-length segmentation, "ρ" represents the language-pair dependent proportion as mentioned in Section 3 "Time (sec.) (Translation\Embedding)" represents time consumption for Translation, which combines data preprocessing and translation process, or Embedding, "Time (sec.) (Similarity)" for "LaBSE (Faiss) + OT" also combines the Faiss search process, when the overlapping rate equals 0.0 representing the fixed-length segmentation without overlapping, and "-" represents for not-used hyper-parameter.

Segment Strategy	Segment Method	Fixed-Length	Overlapping Rate	ρ	F1 Score	Time (sec.) (Translation\Embedding)	Time (sec.) (Similarity)
MT + docalign	web-crawled	-	-	-	0.7880	158.02s	3.93s
LaBSE + Mean-Pool	web-crawled	-	-	-	0.8276	277.29s	0.36s
LaBSE + Mean-Pool	Fixed-Length	150	0.0	-	0.8147	71.17s	0.28s
LaBSE + Mean-Pool	Fixed-Length	150	0.5	-	0.8621	123.96s	0.33s
LaBSE + Mean-Pool	Fixed-Length	200	0.5	0.63	0.8707	123.71s	0.32s
LaBSE + 2 Features	web-crawled	-	-	-	0.8577	291.49s	39.27s
LaBSE + 2 Features	Fixed-Length	200	0.0	-	0.8577	82.44s	38.32s
LaBSE + 2 Features	Fixed-Length	200	0.5	-	0.9009	137.43s	39.92s
LaBSE + 2 Features	Fixed-Length	200	0.5	0.63	0.9009	137.27s	38.89s
LaBSE (Faiss) + OT	web-crawled	-	-	-	0.8362	276.61s	25.92s
LaBSE (Faiss) + OT	Fixed-Length	100	0.0	-	0.8491	69.30s	15.07s
LaBSE (Faiss) + OT	Fixed-Length	100	0.5	-	0.8879	119.44s	16.07s
LaBSE (Faiss) + OT	Fixed-Length	150	0.5	0.63	**0.9224**	121.26s	16.55s