Bilingual Sentence Alignment Using Multilingual Language Models

by

Hovhannes Tamoyan

Physics, Nuclear Reactor Physics, Yerevan State University, 2019

A thesis submitted in partial satisfaction of

the requirements for the degree of

Master of Science

in

Computer & Information Science

in the

COLLEGE OF SCIENCE AND ENGINEERING

of the

AMERICAN UNIVERSITY OF ARMENIA

Supervisor: ____Karen Hambardzumyan_____

Signature: _____ Date:_____

Committee Member: _____

Signature: _____ Date:_____

Committee Member: _____

Signature: _____ Date:_____

Committee Member: _____

Signature: _____ Date:_____

Licenses for Software and Content

Software Copyright License (to be distributed with software developed for masters project)

Content Copyright License (to be included with Technical Report)

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the OC or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License, unless otherwise permitted under applicable Fair Use law.

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the OC, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the OC, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Exceptions are made to this requirement to release modified works free of charge under this license only in compliance with Fair Use law where applicable.

3. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to copy, distribute or modify the OC. These actions are prohibited by law if you do not accept this License. Therefore, by distributing or translating the OC, or by deriving works herefrom, you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or translating the OC.

NO WARRANTY

4. BECAUSE THE OPENCONTENT (OC) IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE OC, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE OC "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK OF USE OF THE OC IS WITH YOU. SHOULD THE OC PROVE FAULTY, INACCURATE, OR OTHERWISE UNACCEPTABLE YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.

5. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MIRROR AND/OR REDISTRIBUTE THE OC AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE OC, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

(This license is known as “OpenContent License (OPL)” and can be found at http://opencontent.org/opl.shtml)

Abstract

In this work, we propose a new algorithm for bilingual sentence alignment based on multilingual language models. We apply it to low-resource languages and low-resource biomedical domains of rich languages and compare it with the existing alignment algorithms. We have developed machine translation systems for the WMT20 Biomedical Text translation task for four translation directions and achieved the highest BLEU scores for two of them. We further show the advantage of XLM-R (a completely unsupervised multilingual language model) over LASER (a sentence embedding algorithm that requires some labeled data) for parallel corpus alignment of low-resource languages.

Introduction

This work represents our work on data filtering, alignment, and processing pipeline intended to improve the quality of parallel data used for training state-of-the-art neural machine translation systems. We discuss various parts of the pipeline that can hugely impact the performance of such models. In particular, we focus on the bilingual sentence alignment problem by proposing a new algorithm, performing extensive experiments, comparing with the prior work, and showing the effectiveness of our methods, especially for low-resource languages.

This paper includes YerevaNN's Neural Machine Translation system and the developed pipeline for the WMT20 Biomedical Translation task. The pipeline comprises the multilingual language model-based parallel sentence preprocessing toolkit (parasite), which aligns bilingual corpora resulting in higher BLEU scores on machine translation task. Provided systems are for English↔Russian and English↔German language pairs. Our submissions achieve the best BLEU scores for the English Russian pair, with the English→Russian direction outperforming the other systems by a significant margin (Hambardzumyan et al., 2020; Bawden et al., 2020). The submitted translation systems are pioneering for all the submitted pairs in the human evaluation stage.

We propose a new algorithm for bilingual sentence alignment, which helps to drastically improve the quality of the corpora and the NMT system. We further compare this method with already known ones. As a continuation of the bilingual corpora alignment problem, we discuss primarily used algorithms: Vecalign, Bleualign (NMT), Hunalign w/ lexicon, etc., Present the time complexity and alignment scores by showing the significance of Vecalign over the other algorithms on a given set. We examine the choice of the multilingual model for bilingual sentence alignment by comparing the two pre-trained models. Moreover, we use Vecaling as the main algorithm for future experiments. We compare alignment F_1 scores using LASER and XLM-R pre-trained models embeddings' on German↔French, English↔Kazakh, and Armenian↔Kazakh language pairs. We show the advantage of XLM-R, a completely

unsupervised multilingual language model on low-resource languages bilingual sentence alignment tasks over LASER, a supervised multilingual model. Provide PCA visualizations of different combinations of languages on a given multilingual model, plus pairwise distance calculation results. Afterward, training Machine Translation systems to spotlight the obtained results' significance on the downstream task.

The work is organized as follows: Section 1 discusses related works and state-of-the-art architectures, Section 2 describes WMT20 Biomedical Translation task submissions, pipeline, and the new bilingual sentence alignment algorithm: "Parasite," experiment with Section 3 presents the comparison of multilingual models for alignment algorithms, Vecalign's comparison with other alignment algorithms, obtained F_1 scores, PCA visualizations, distance matrix construction, their average distance measurements, and downstream task results.

Finally, the Appendix presents extended versions of the illustrations of PCA and distance matrices, along with more tables with experimental results.

1 Related Work

1.1 WMT20 Biomedical Machine Translation

WMT is a Conference on Machine Translation that is conducted every year, with multiple shared tasks. The proposed models combine newly suggested architectures and techniques to achieve the highest automatic and human evaluation results every year. One of the proposed shared tasks is a low-resource domain translation task: Biomedical Text translation. English↔Russian direction was new for WMT20 Biomedical Translation; meanwhile, in English↔German direction, two teams ARC (Peng and Liu, 2019) and UCAM (Saunders and Stahlberg, 2019) recorded almost identical BLEU scores. Researchers in these two teams focused on transfer learning methods or attempted to mix the training data with other sources to address the data scarcity issue.

Mainly UCAM tackled the problem by using an ensemble model. They have applied transfer learning iteratively on datasets from different domains, obtaining strong models covering two domains for the English↔German language pair. They have used a general news translation model to do transfer-learning on an in-domain dataset, "UFAL Medical." Meanwhile, ARC, except making domain adaptation on a pre-trained model, also trained and fine-tuned their systems from scratch on a mix of out-of-domain and in-domain parallel corpora as well. They admit that enhanced performance is attributed to more in-domain training. Eventually, their fine-tuned models scored higher BLEU scores compared with the ones trained from anon. Both articles admit that their data preprocessing pipelines play a significant role in results quality enhancements.

ARC removes potentially misaligned sentences based on “fast-align” scores when it comes to misaligned sentences, which at first glance seems a considerable loss. UCAM does not mention the problem of misaligned sentences, so they use the dataset as it is.
The most common architecture used in NMT systems is the Transformer described in (“Attention Is All You Need,” Vaswani et al., 2017) and its extended Transformer Big modification, which we will cover more in-depth in the following section.

1.2 Bilingual Sentence Alignment

In the early days of statistical MT, sentence alignment was a central research topic but gradually received less attention once standard sentence-aligned parallel corpora became available (Thompson and Koehn, 2019). Interest in low-resource MT has led to a renewal in data gathering methods (Buck and Koehn, 2016; Zweigenbaum et al., 2018; Koehn et al., 2019); however, we find little recent work on bilingual sentence alignment. One of the leading papers that wraps a significant part of the work done in bilingual sentence alignment is presented by (Thompson and Koehn, 2019) in “Vecalign: Improved Sentence Alignment in Linear Time and Space” paper. In this paper, the authors use a multilingual model LASER to obtain the sentence embeddings; while publishing the paper, the LASER was the State Of The Art (SOTA) multilingual model so far. That’s how the use of LASER is justified. Just a few months after that, mBERT and XLM-R came to light. LASER is a BiLSTM based architecture; meanwhile, mBERT and XLM-R are Transformers. However, they essentially differ in architecture, but they do the same job in general; they are multilingual models, which means their objective is to create a latent space where sentences’ semantics will be closer despite their languages.

Vecalign is the pioneering algorithm for bilingual sentence alignment in high and low-resource settings and improves MT quality (Thompson and Koehn, 2019). First, it has the lowest running time complexity; secondly, it outperforms all algorithms in terms of binary classification test accuracy score: F_1 (Thompson and Koehn, 2019).
In general, there are three main approaches to the problem of text alignment at the sentence level: length-based (Brown et al. 1991, Gale and Church 1991), dictionary or translation based (Chen 1993, Melamed 1996, Moore 2002), and partial similarity-based (Simard and Plamondon 1998). This prior method may work well for a family of languages such as German and English.

As a low-resource language pair, we consider Kazakh in our experiments of XLM-R and LASER comparison. As for the WMT19, a news translation task was proposed: a low-resource translation task on English↔Kazakh direction. For English→Kazakh direction, there were 13 submitted models and 11 on Kazakh→English (Barrault et al., 2019). All the submitted systems did not outperform human translations. Most of the proposed models were based on back-translation or pivoting techniques; some used Russian or Turkish as a third language to

translate a given sentence from Kazakh to Russian/Turkish than to English. The choice of the third languages is based on alphabetical similarities and family dependence.

On Kazakh→English direction, the system proposed from the “NEU” team scored the highest results on the human evaluation stage, “rug-morfessor” followed them with a modest difference of score. In English→Kazakh, the leading group was the “UAlacant-NMT” with its two systems, then the “NEU” team follows. Because (“Findings of the 2019 Conference on Machine Translation (WMT19)” Barrault et al. 2019) do not report BLEU scores for English↔Kazakh directions, but human evaluation results; we base on the papers reported BLEU scores individually. It should be noted that that the presented results in each paper were not explicitly represented, in the sense that in some papers, the evaluation sets were not mentioned; probably, most of them were focused on their methods comparison on local sets. However, as the NEU claims, the team achieved the highest score in English↔Kazakh directions (Li et al., 2019). They have recorded poor results by doing vanilla training; meanwhile, significant improvements are seen when applying back-translation and pivoting techniques.

2 WMT20 Biomedical Machine Translation

2.1 Introduction

WMT20 Biomedical Translation Task is a narrow in-domain Machine Translation (MT) problem (Bawden et al., 2020). In such tasks, the available bilingual data is limited and often noisy, which generates many challenges. The proposed directions for this particular task were 15. We have participated in English↔Russian and English↔German language pairs Biomedical Translation task by scoring considerable results. We explain our success by heavy data preprocessing, especially by parallel corpora alignment. Alignment errors have been noted to have a negligible effect on statistical MT performance (Goutte et al., 2012). However, misaligned sentences are much more detrimental to neural MT (NMT) (Khayrallah and Koehn, 2018).

The choice of the aforementioned 4 directions is explained by time-resource limitations and the author’s knowledge of languages. So far, the automatic measures used for NMT quality scoring do not fully replicate translation fluency and semantics but score the corresponding words matching. The knowledge of the languages is preferred to do an approximate human evaluation on development stages, and as well analyze the dataset.

We have manually cleaned and aligned a much higher quality subset of the original “MEDLINE” training data for local evaluation purposes. The goal of this manual analysis is to have a reliable test set for the evaluation of the proposed methods. This manual step is useful

only during the development of the algorithm. It will not be required in subsequent applications of our methods.

The insights collected during this manual analysis were then used to fix the most common issues within the training data. In particular, we noticed that the original dataset contained paper abstracts in two languages without sentence-level alignments. The organizers' training corpus was created using an automated sentence segmentation and alignment process, which failed most of the time. We built a data pipeline that handles cleanup, sentence segmentation, alignment of translation sentence pairs, and preprocessing. Performed operations recorded a significant boost on the quality of NMT in terms of BLEU score.

2.2 Architecture

All our NMT models are built on top of WMT19 News Translation task winner models by (Ng et al., 2019). We employ the FairSeq library (Ott et al., 2019) to fine-tune pre-trained models on the in-domain translation data. It is worth mentioning that this model is a sequence to sequence model; this being said, the translation (or any other task) is done on the sentence level and not on the document level. From this, it is ideologically obvious that high-quality, aligned parallel data is necessary for good quality results.

The used architecture is Transformer: a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. The Transformer allows for significantly more parallelization. The Transformer model extracts features for each token using a self-attention mechanism to determine how important all the other words in the sentence are concerning the preceding word. Moreover, no recurrent units are used to obtain these features; they are weighted sums and activations that are parallelizable and efficient. The schema of the Transformer is presented in Figure 1.

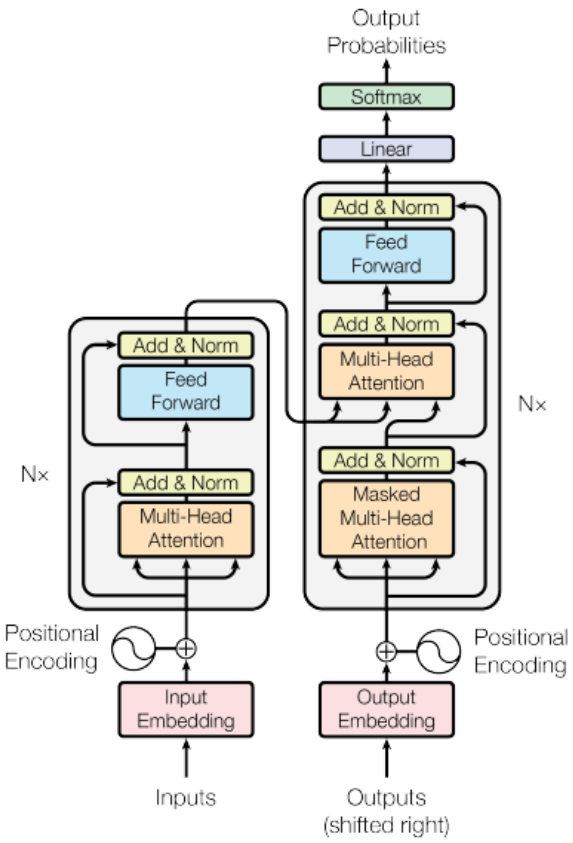

Figure 1: Architecture of Transformer. The figure is taken from ("Attention is all you need" Vaswani et al., 2017).

We can observe an encoder structure on the left side and the decoder on the right. Both contain a core block of "an attention and a feed-forward network" repeated n times.

The core concept of the self-attention mechanism is the following: a sequence-to-sequence operation, if we denote the input vectors x_1, x_2,…, x_t and the corresponding output vectors y_1, y_2,…, y_t. The vectors all have dimension k. To produce output vector y_i, the self-attention operation takes a weighted average over all the input x vectors, and the most straightforward option is to calculate the dot product of those. The three critical elements for self-attention are Queries, Values, and Keys. For every encoder input (token embedding), the 3 vectors are created. These vectors are created by multiplying the embedding by three matrices that we trained during the training process. The second step in calculating self-attention is to calculate a score. The aim is to score each token of the input sentence against this token. The score determines how much focus on putting other parts of the input sentence as we encode a word at a particular position. It is calculated by taking the dot product of the query vector with the key vector of the respective token.

The pre-trained models used for our experiments are based on the "transformer_wmt_en_de_big" architecture. In the previous works, the same Transformer architecture is used with a modified feedforward network dimension (8192) and a shared matrix for input(encoding) and output(decoding) embeddings.

We start fine-tuning single_model versions of Facebook's WMT19 models on in-domain (biomedical) parallel data and stop the training when the perplexity on the validation set does not improve for 5 consecutive epochs. To fight noisy training data, we use labelsmoothed cross-entropy loss (Müller et al., 2019). Also, because we are fine-tuning, the neural architecture and related implementation details cannot be changed in the fine-tuning scenario. While this limited our experimental setup, it also allowed us to care less about hyperparameter tuning and focus on other parts of the pipeline.

2.3 Parallel Data

For all directions, we use only "MEDLINE" training data provided by the shared task organizers. We chose 50 random documents from the training data as the validation set. In the case of English↔German (en↔de), we use "OK"-tagged sentence-pairs from the WMT19 Biomedical Translation test set (Bawden et al., 2019) as the local test set, which is manually aligned. To have a local test set of similar quality for English↔Russian (en↔ru), we take another random 50 documents, then manually fix misaligned sentences and filter out a few pairs with incorrect translations. Eventually having a dataset which consists of: training set of 47,745 (en↔ru) and 37,469 (en↔de), development or validation set of 425 (en↔ru) and 652 (en↔de), and test set of 553 (en→ru), 463 (ru→en) and 783 (en→de) and 612 (de→en) sentences.
We noticed that the provided data was poorly aligned during the manual review of the en↔ru local test set. It was possible to get high-quality sentence pairs by re-aligning the sentences (only 9 sentences were dropped except the titles/subtitles, out of 504 sentences). Then we tried to use these insights to build a new automated system for monotonic alignment of the sentences (described in Section 2.4).

Table 1 exhibits a document containing the most common issues found during manual analysis in the MEDLINE training data: the bitext documents may be misaligned: the translation of a source sentence may appear on a different line, or even on multiple lines, on the target side, headings and section names may occur next to a sentence on one side only, or on both sides, English documents may start with titles (often wrapped in brackets), while the Russian ones do not.

1	*[Risk factors of stroke in men exposed to environmental factors at workplace]*. OBJECTIVE	Цель исследования - изучение факторов риска развития инсульта у мужчин разных возрастных групп, подвергающихся воздействию неблагоприятных производственных факторов
2	To explore risk factors of stroke in men of different age groups exposed to adverse environmental factors at work.	Материал и методы
3	MATERIAL AND METHODS Four hundred and eleven men after stroke, aged from 30 to 65 years, including 335 patients, who had been exposed to adverse environmental factors at work, were compared to 76 patients who had not been exposed to adverse environmental factors.	Обследованы 411 мужчин в возрасте от 30 до 65 лет, перенесших инсульт, из них 335 пациентов подвергались влиянию неблагоприятных производственных факторов и 76 пациентов, которые воздействия вредных факторов не испытывали (группа сравнения).
4	RESULTS	Результаты.
5	The distribution of the frequencies of risk factors of stroke depending on the character of adverse factors was shown.	Установлена частота распределения факторов риска развития инсульта у мужчин в зависимости от характера профессиональных вредностей.

Table 1: A sample document (id 26978637) from the MEDLINE en↔ru training set exhibits the most common issues in the dataset. The first line in English includes the title of the paper, which is not present in Russian. The English version of the main content of the first line in Russian is given on the second line. In Line 3, English has an extra heading that corresponds to Line 2 on the right side. The rest of the third line on the left matches the third line on the right side, and the last two lines are perfectly matched.

These issues are too common in the training set, and simply removing incorrect pairs of sentences would significantly reduce the dataset. Instead, we decided to fix the misaligned sentences to preserve as much parallel content as possible. The solution is described in the following section.

2.4 Monotonic Alignment

The problems of the training set described in Section 2.3 can be caused by poor: XML parsing, sentence segmentation, or monotonic segment alignment method. Here we describe a novel method for monotonic sentence alignment using multilingual language models and discuss the contribution of its hyperparameter choices. Multilingual language models have

been previously shown to be effective in parallel data mining (Kvapilíková et al., 2020). We also compare our approach to the baseline data pipeline provided by the shared task organizers based on the Syntok segmentation system and GMA (Melamed, 2001).

Our method of monotonic sentence alignments is as follows: we calculate a similarity matrix of all source-target candidate pairs and decode pairs to maximize the similarity of the resulting sentence pairs. We consider two approaches for the decoding step: greedy and dynamic.

The similarity matrix is calculated using Euclidean and Cosine distances of sentence embeddings from a pre-trained multilingual language model. We found xlm-roberta-large (Conneau et al., 2019) to be the best one. In order to obtain a fixed-size vector for each sentence, we simply take the average of the wordpiece embeddings (Cer et al., 2018; Artetxe and Schwenk, 2019). The choice of taking the average of the token embeddings is justified not only by our experiments but also by (Cer et al., 2018; Artetxe and Schwenk, 2019) observations. We experimentally see that taking the average of the embeddings results in the best representation of a sentence (Appendix Table 4). We get the highest alignment F_1 scores using this technique. One might consider a Sentence Transformer choice; however, finding a pre-trained one for a given language pair is not an easy task, and training it from zero is expensive. Meanwhile, our algorithms focus on cheap and at the same time effective approaches. The similarity matrix for the document in Table 1 is presented in Table 2.

	Цель иссл ... факторов.	Материал и методы.	Обследованы ... сравнения).	Результаты.	Установлена ... вредностей.
[Risk factors ... workplace].					
OBJECTIVE					
To explore ... at work.					
MATERIAL AND METHODS					
Four hundred ... factors.					
RESULTS					
The distribution ... shown.					

Table 2: A sample document (id 26978637) from the MEDLINE en↔ru training set (Table 1). XLM-R is used to compute the contextual word embeddings. The sentence embedding is considered as the average of all the word-piece embeddings. The similarity matrix is calculated using Euclidean distance, pairwise between all the source and target segments. Note: the darker the color, the higher the similarity score, thus the darker, the better. The outlined rectangles indicate the rectangles/correspondences to be chosen.

We assume that translation pairs are distributed in a monotonic way in the document: without cross-matching. Using the greedy/dynamic approach, we decode the best path with the highest total similarity. We also use multi-sentence windows as possible translation candidates to support one-to-many mappings.

We also attempt to address some common issues concerning the given MEDLINE abstracts that may harm the quality of the alignments:

1. remove titles from the English version that are absent in the Russian version,
2. detect the headings that often get attached to adjacent sentences,
3. lowercase the text before obtaining embeddings (as the English headings are written in capitals, unlike the Russian ones),
4. experiment with different sentence segmentation systems such as SciSpacy (Neumann et al., 2019) (in-domain, for English) and Razdel3 (for Russian),
5. penalize candidates with source/target length ratios exceeding 2.

Additionally, we consider using normalized distances and the margin-based approach described in (Artetxe and Schwenk, 2019).

The parasite works in an end-to-end manner: it takes the documents to be aligned, flags regarding text preprocessing, and outputs the aligned documents. One of the flags is designed for sentence segmentation choice, which can be different depending on the language. The rest of the flags are language-independent, and thus the overall algorithm is language independent.

2.4.1 Greedy Approach

In the greedy approach, we construct the set of correct sentence pairs in an iterative process. Given the similarity matrix, we add the sentence pair with the maximum similarity score at each step. There is an assumption that the alignments should be monotonic; after each step, we exclude all remaining candidate sentence pairs that would break the monotonicity. Monotonicity in this context means that crossing matches of alignments are excluded. Our implementation finds at most one target sentence for a source sentence (and vice versa), so the window size is 1.

Algorithm 2: Greedy decoding

S_N , T_M ← source and target sentences
$D_{i,j}$ ← Sim(S_i , T_j)
Res ← {}
while |Align| < min(N, M) **do**
    i, j ← arg max(D)
    Align ← Align ∪ {i, j}
    $D_{i..N,0..j}$, $D_{0..i,j..M}$ ← 0
end
Result: Align

2.4.1 Dynamic Approach

In the dynamic algorithm, we consider maximizing the sum of the similarity scores of the selected sentence pairs according to the given matrix. Our implementation of this approach, unlike the greedy one, can produce sentences consisting of multiple (up to K) segments on each side. To find the mapping with the best total similarity score, we use dynamic programming.

Algorithm 2: One-to-many (K) dynamic decoding

```
S_N , T_M ← source and target sentences
Best_{N,M} ← 0
Res_{N,M} ← {}
for i = 1 → N do
    for j = 1 → M do
        for u, v = 1 → K do
            candidate ← Best_{i-u, j-v} + Sim(S_{i-u..i}, T_{j-v..j})
            if candidate > Best_{i,j} then
                Best_{i,j} ← candidate
                Res_{i,j} ← Res_{i-k, j} ∪ {S_{i-k..i}, T_{j-v..j}}
            end
        end
    end
end
```

2.5 Results

We prepared three submissions for the WMT20 Biomedical Translation Task: run1 for all directions was the baseline model. At the same time, for run2 and run3, we chose the best models according to their BLEU score on the local test set at the time of the submission. In run2 and run3, all the models besides de→en of run2 are trained with our data pipeline and bigger batches.

We also applied the back-translation technique to improve NMT system quality. We obtain translations with “wmt19.en-de.joined-dict.single_model”, “wmt19.de-en.joined-dict.single_model”, “wmt19.en-ru.single_model” and “wmt19.ru-en.single_model” pre-trained models. Then fine-tune new models on a mixed data consisting of the regular parallel training data (biomedical) and back-translated data with equal proportions.

To perform back-translation, we need a set of in-domain monolingual sentences that do not overlap with the test set. To train back-translated de↔en and ru↔en directions, we took all English sentences from all parallel corpora available from previous years’ MEDLINE set (both

training and test sets), excluding the parallel corpora we would eventually train on. This way, we collected 296,052 (236,379) English sentences for German (for Russian). To obtain a parallel corpus, we translated them using our models and then filtered them using the same process as with the regular training data. Eventually we had 281,054 (220,916) sentence pairs for en↔de (en↔ru).

We did not perform back-translation from Russian or German (directions en→de and en→ru), as we did not expect to find in-domain sentences that are not present in MEDLINE. For de→en of run2, back-translation data was collected with beam search (size of 8), in case of ru→en, we had noise added similar to (Edunov et al. 2018) they showed that adding input-level noise significantly boosts the back-translation performance, and for run3, we used a simple sampling strategy. Our experiments with back-translation showed no significant advantage of any of those compared to the others. For run2 and run3, we used v2 preprocessing; the sentence splitting was done with scispacy (for English and German) and a slightly modified version of razdel (for Russian).

After our submissions, we further improved our data pipeline. Table 3 is an empirical analysis of the effect of different components of our data pipeline, as measured by the performance on the final translation task.

Model	**en→ru**	**ru→en**
baseline model	27.7	30.7
+ v2 preprocessing	30.5	31.3
+ train with bigger batches	30.7	31.3
+ greedy alignments	30.1	31.8
+ detect section names	30.7	32.3
+ remove titles	31.3	**32.5**
+ optimize total similarity	30.4	32.2
+ normalize distance matrix	30.8	32.1
+ penalize source/target ratio	31.2	31.5
+ one-to-many (K=3)	**32.2**	32.3
Vecalign (LASER) (K=3)	29.0	29.6
Vecalign (XLM-R) (K=3)	27.6	29.1

Table 3: The effect of different components of the data processing pipeline. Given BLEU outcomes are scored on the local test set.

Each row of the table corresponds to a model trained on the data obtained from a pipeline with specific components enabled. All the models are trained by fine-tuning the general domain baseline using our default hyperparameters. We measure the BLEU score on the local test set.

We noticed several issues with our preprocessing pipeline during our early experiments, which we fixed for the later experiments. In particular, we noticed that some sacremoses command-line flags were broken, and the out-of-the-box inference tool from FairSeq did not fully replicate the preprocessing pipeline used for training (punctuation normalization and vocabulary-aware subword segmentation). The original pipeline (called v1) was used for our baseline models. The later experiments used the fixed implementations of the sacremoses and FairSeq (denoted by v2).

Fixing the issues of the standard preprocessing (v2 vs. v1) gives a significant boost, especially when decoding to Russian (en→ru direction). The effect of training with bigger batch sizes gives only a slight improvement, while the absolute training duration reduces drastically. As mentioned previously, there were issues with section names and titles in the provided parsed documents. After addressing these issues, our greedy approach gives better

alignments. The total similarity optimization using dynamic programming is not always better than the greedy method, but the performance improves en→ru with another +1.1% BLEU score. Overall, the new data pipeline enhances NMT performance: +1.6% BLEU for ru→en and a more considerable gain of +4.5% BLEU score for en→ru. Although we observe consistent performance improvement for both directions en↔ru, the effect for en→ru direction is more significant. We could not determine the reason for such asymmetry.

In our experiments, we solely used the MEDLINE dataset given by WMT20 biomedical translation task organizers. We chose our baseline model and two other models with the highest BLEU scores on a local test set as our three submissions. The best ones got 35.2% BLEU on English-German and 41.3% on German-English test sets. For English-Russian and Russian-English directions, we reached BLEU scores of 37.9% and 43.2%, respectively, the best scores among all submissions of the WMT20 Biomedical Translation Task.

Our system scores the best results for human evaluation for en→ru direction than all submitted systems. However, reference sentences still achieve better results. Almost the same picture is for the opposite direction (ru→en). Meanwhile, we outperform one of the teams and the reference translation in en→de direction, having an almost equal performance with another two, losing a team. In de→en direction performing better than a team, and having nearly the same results with others.

3 Bilingual Sentence Alignment

3.1 Introduction

This section discusses the need for good-aligned parallel data for the NMT task and examines available algorithms for bilingual sentence alignment. We consider using the XLM-R pre-trained model instead of LASER. We measure the pros and cons of two models on the Vecalign bilingual sentence alignment algorithm, mainly focusing on low-resource language pairs data alignment. Thompson and Koehn considered aligning low-resource data but did not experiment with using a multilingual model other than LASER; they note that the algorithm is not LASER specific (Thompson and Koehn, 2019). The choice of XLM-R boils down to the problem of available parallel data. As we will discuss in detail in (Section 3.3), the LASER is trained on parallel data; meanwhile, in XLM-R, monolingual data is being used. This being said, LASER is a supervised and XLM-R an unsupervised language model. This structural difference yields a well-separated choice. When working on low-resource language or low-resource in domain but rich language translation tasks, the available parallel data is very scarce, and the better option, in this case, is to go with a model which does not require bilingual data; such as XLM-R.

3.2 Vecalign

Vecalign algorithm proposes a novel sentence alignment scoring function based on the similarity of bilingual sentence embeddings (Thompson and Koehn, 2019). Embeddings of blocks of sentences are obtained by averaging sentence embeddings produced by a multilingual model. The size of the resulting vector does not depend on the size and number of sentences; thus, the time complexity of comparing the similarity of blocks of sentences does not depend on the number of sentences being compared.

In Vecalign, they show empirically that average embeddings for blocks of sentences are sufficient to produce approximate alignments, even in low-resource languages. This enables them to approximate Dynamic Programming (DP) in O(N+M) in time and space.

The similarity between sentence embeddings is used as the scoring function for sentence alignment (Thompson and Koehn, 2019). Sentence embedding similarity has been shown to effectively filter out non-parallel sentences (Hassan et al., 2018; Chaudhary et al., 2019) and locating parallel sentences in bilingual corpora (Guo et al., 2018).
For similarity score calculation, cosine distance is an obvious choice for comparing embeddings. Still, it has been noted to be globally inconsistent due to “hubness” (the tendency of high-dimensional data to contain hubs/groups) (Radovanovic´ et al., 2010; Lazaridou et al., 2015). High-dimensional spaces are often affected by hubness (Radovanovic´ et al., 2010). The hubness is the tendency of some vectors “hubs” to appear in the top neighbor lists of many sample items: that is, they contain certain elements – hubs – that are near many other points in space without being similar to the latter in any meaningful way (Lazaridou et al., 2015). Thompson and Koehn (2019) propose normalizing with randomly selected embeddings as it has linear complexity. Sentence alignment seeks minimal parallel units, but they find that DP with cosine distance favors many-to-many alignments. They scale the cost by the number of source and target sentences considered in a given alignment to remedy this issue. And the resulting scoring cost function is:

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

where x, y denote one or more sequential sentences from the source/target document; cos(x,y) is the cosine similarity between embeddings of x, y; nSents(x), 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.

Instead of searching all possible alignments via dynamic programming, Vecalign suggests half the size of the sentences in the source and the target by averaging them, which will produce $(\frac{N}{2})(\frac{M}{2})$cost for comparisons. Then, they refine this approximate alignment using the original sentence vectors, constraining to a small window around the approximate

alignment. A window size ω must search, which at a minimum should be large enough to consider all paths covered by the lower-resolution alignment path, but also can be increased to allow recovery from minor errors in the approximate alignment. The length of the refinement path to search is at most $(N + M)$ (deletions/insertions), so refining the path requires at most $(N + M)\,\omega$ comparisons. Thus the whole NM comparisons can be approximated by $(N + M)\,\omega + (\frac{N}{2})(\frac{M}{2})$ comparisons. Applying this logic recursively, the quadratic NMcost can be approximated with a sum of linear costs:

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

Considered operations are insertions, deletions, and 1–1 alignments in all but the last search. Recursive downsampling and refining of DP was proposed for dynamic time warping in Salvador and Chan (2007) but has not previously been applied to sentence alignment.

So far, known alignment algorithms are 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).

The second recently released alignment algorithm is “Bleualign”: a sentence alignment algorithm that, instead of computing an alignment between the source and target text directly, bases its alignment search on an MT translation of the source text. The main disadvantage of an MT-based algorithm is that it requires an existing MT system. For low-resource language pairs, this requirement makes the algorithm unattractive because the process of obtaining an aligned data and an adequate MT falls into a loop. Bleualign initially used an SMT for MT, but the algorithm is not limited to that, so an NMT can be used to boost performance (Sennrich and Volk, 2010). Another method: “Coverage-Based,” is based on Bleualign; their scoring function uses Moses phrase tables by which they avoid the need for an MT system (Gomes and Lopes, 2016).

“Hunalign” is the primary third used one. It concentrates on dictionary and length-based methods and gives a hybrid algorithm. In the first step of the algorithm, a raw translation of the source text is produced by converting each word token into the dictionary translation with the highest frequency in the target corpus or to itself in case of lookup failure. Next, an IBM ’Model I’ translation model (Brown et al. 1991) is trained on the set obtained from the first phase. Third, the similarity is calculated using this translation model, combined with sentence length similarity. The output alignment is calculated using a custom score. Hunalign can be used in two “ways,” one with a lexicon/dictionary and the second without it, so if a dictionary does not exist, the algorithm can build one by traversing through the set (Varga et al., 2007).

Some experiments are done on manually aligned yearbook articles published in multiple languages by the Swiss Alpine Club from the Text+Berg corpus, and this dataset is being released with Bleualign paper. Alignment F_1 scores are calculated on the previously mentioned algorithms. The results are shown in Table 4 (Thompson and Koehn, 2019).

Algorithm	**O ()**	**P**	**R**	**F1**
Gargantua	N^2	0.48	0.54	0.51
Hunalign w/o lexicon	**N**	0.59	0.70	0.64
Hunalign w/ lexicon	**N**	0.61	0.73	0.66
Gale and Church (1993) †	N^2	0.71	0.72	0.72
Moore (2000) †	‡	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	**N**	**0.89**	**0.90**	**0.90**
Vecalign (XLM-R)	**N**	0.75	0.78	0.76

Table 4: * Best reported in Gomes and Lopes (2016). † Best reported in Sennrich and Volk (2010). O(‡) data-dependent time complexity (Thompson and Koehn, 2019).

As we can see, the Vecalign algorithm outperforms the second best method by 0.5 F_1 points and performs worse with XLM-R. Bleualign improves its results by using an NMT instead of an SMT. Hunalign and Vecaling provide linear time complexity; meanwhile, Bleualign, Coverage-Based, Gale and Church, and Gargantua methods are quadratic. So from time complexity perspective as well, Vecalign outperforms the others.

3.3 LASER Versus XLM-R

LASER is an architecture to learn joint multilingual sentence representations for 93 languages, belonging to more than 30 different families and written in 28 various scripts; it uses a single BiLSTM encoder with a shared BPE vocabulary for all languages, which is coupled with an auxiliary decoder and trained on publicly available parallel corpora (Artetxe and Schwenk, 2019).

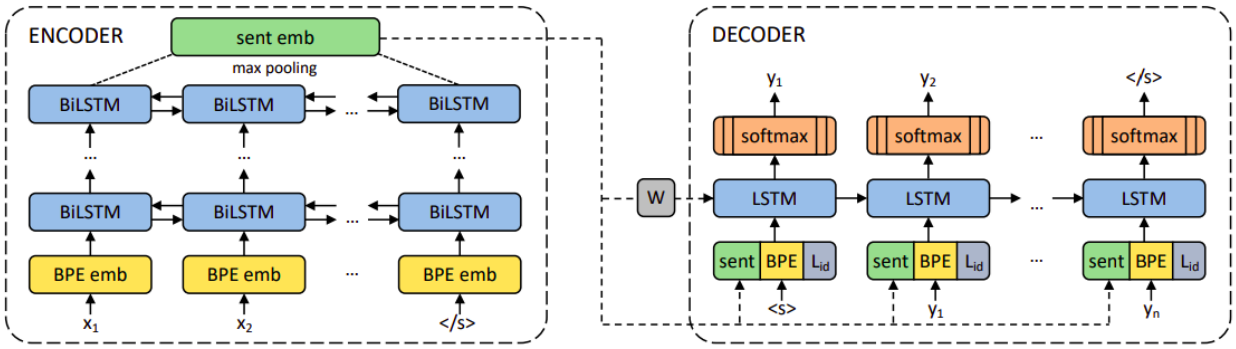

Figure 2: Architecture of LASER system to learn multilingual sentence embeddings. Based on multiple BiLSTMs. The figure is taken from ("Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond" Artetxe and Schwenk, 2019)

As shown in Figure 2, LASER consists of two major parts: the encoder and the decoder; in the encoder, BPE embeddings are learned. The resulting matrices are passed to multiple numbers of layers of BiLSTM. Sentence embeddings are obtained by applying a max-pooling operation over the output of a BiLSTM encoder. The obtained sentence embeddings are used to initialize the decoder LSTM through a linear transformation and are also concatenated to its input embeddings at every time step. We can see that no other connection between the encoder and the decoder exists, as we want all relevant information of the input sequence to be captured by the sentence embedding.

The joint byte-pair encoding (BPE) vocabulary is used with 50k tokens, which is learned on the concatenation of all training corpora. This way, the encoder has no explicit signal on the input language, encouraging it to learn language-independent (multilingual) representations. In reverse, the decoder takes a language ID embedding that specifies the language to generate, which is concatenated to the input and sentence embeddings at every time step. The list of the languages and number of sentences respectively used can be found in the paper of LASER (Artetxe and Schwenk, 2019).

XLM-R is a Transformer based masked language model trained on 100 languages, using 2TB of filtered CommonCrawl data (Conneau et al., 2020). As the authors report, XLM-R significantly outperforms multilingual BERT(mBERT) on various cross-lingual benchmarks. XLM-R heritages the architecture of XLM, only making slight modifications that improve performance at scale. The used architecture is the Transformer model (Vaswani et al., 2017), trained with multilingual MLM objectives using only monolingual data. The approach randomly picks streams of text from each language and trains the model to predict the masked tokens in the input. Sentence Piece subwork tokenization is used (Kudo and Richardson, 2018), and the used vocabulary has 250K tokens. Table 5 presents the languages used in this work and the number of sentences on each model training set.

Model	fr (French)	de (German)	en (English)	kk (Kazakh)	hy (Armenian)
LASER (sentences)	8.8M	8.7M	2.6M	4k	6k
XLM-R (tokens)	9.7k	10.2k	55.6k	476	421

Table 5: Number of sentences used on LASER and XLM-R for a given language. Note that XLM-R gives the number of tokens which is approximated to sentences in the braces.

(Artetxe and Schwenk, 2019) present the number of sentences and (Conneau et al., 2020) the number of tokens, so assuming that a sentence might contain a token, they still drastically differ.

Of course, it is not just the size that matters, but the quality of the datasets; thus, it is noteworthy that the dataset used for XLM-R is clean "CommonCrawl" (Conneau et al., 2020), for LASER "Europarl," "United Nations," "OpenSubtitles2018", "Global Voices," "Tanzil," and "Tatoeba" corpora are combined to form the final dataset (Artetxe and Schwenk, 2019). The dataset used for LASER is not parallel in all directions (not all pairs exist in the corpus); they picked bitexts aligned with two target languages, choosing English and Spanish for that purpose.

3.3.1 Principal Component Analysis

Multilinguality in a broader sense means the model's ability to group a given set of sentences through semantics rather than languages. To get a visual understanding of how multilingual LASER and XLM-R are, we present Principal Component Analysis (PCA) graphs. As a "benchmark" representation, we exhibit Sentence Transformers results on the same set (Reimers and Gurevych, 2019). Sentence Transformers have the same structure as the regular Transformers, and they train the model by giving sentences as input. Unfortunately, there are not so many pre-trained sentence language models available out there. Usually, the existing ones are trained in few languages (2 or 3). We use T-systems already trained models taken from their huggingface page. The data used to visualize are taken from huggingface's provided dataset package, from the "XNLI" problem's train set by randomly picking 1k sentences to train the PCA components on and 12 sentences to visualize.

Plots in Figure 3 represent that Sentence Transformer is making clusters on a sentence level as well does LASER; meanwhile, XLM-R is clustering on a language level, which is undesirable for multi-linguality. This can be seen when no center zero/normalize operation is applied; the clusters are apparent. Nevertheless, when centering zero, the embeddings language level clusters break, and the overall picture tends to form semantic level clusters

instead. The whole point of multilingual embedding is that its embeddings do not distinguish between languages but share common vector space.

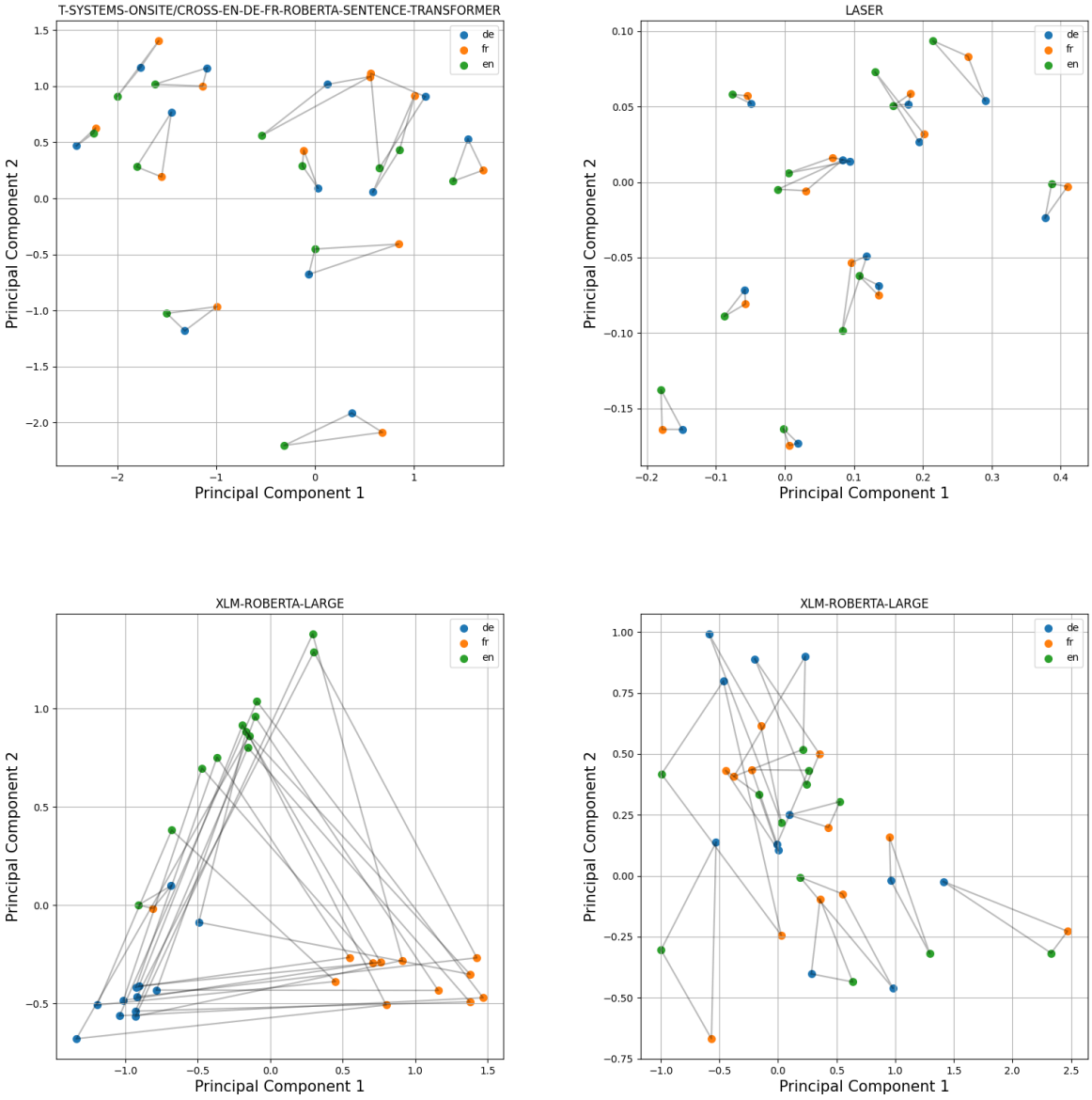

Figure 3: Principal Component Analysis of embeddings extracted from Sentence Transformer, LASER, and XLM-R on 3 languages (en, de, fr). Sentences are taken randomly from huggingfaces “datasets” -> “xnli” -> “train” -> ”premise”. 1200 sentences for training the PCA model, and 12 to visualize. Top left: “T-Systems Sentence Transformer” trained on en, de, fr languages, top right: LASER, bottom left: XLM-R with mean pooling, bottom right: XLM-R with mean pooling and embeddings centered zero.

From Plots in Appendix Figure 1 can be seen that; XLM-R with mean pooling makes clear clusters of languages, even showing the closeness of kk (Kazakh) sentences in two

different corpora, en (English), fr (French), and de (German) lay closer, and hy (Armenian) is far away from all the languages because of its structural and alphabetical differences. Things are relatively not straightforward in LASER's PCA; not clear clusters on language level nor semantic. For XLM-R, after applying center zero operation, they make significant groups of semantics, but noisy things prevail.

After all the resulting PCA's, it is evident that center zero operation decreases the distance between a pair of sentences by promoting multilingualism on average. For high-resource languages (de, fr, en), multilingualism persists with LASER extracted embeddings and diminishes for low-resource languages (hy, kk).

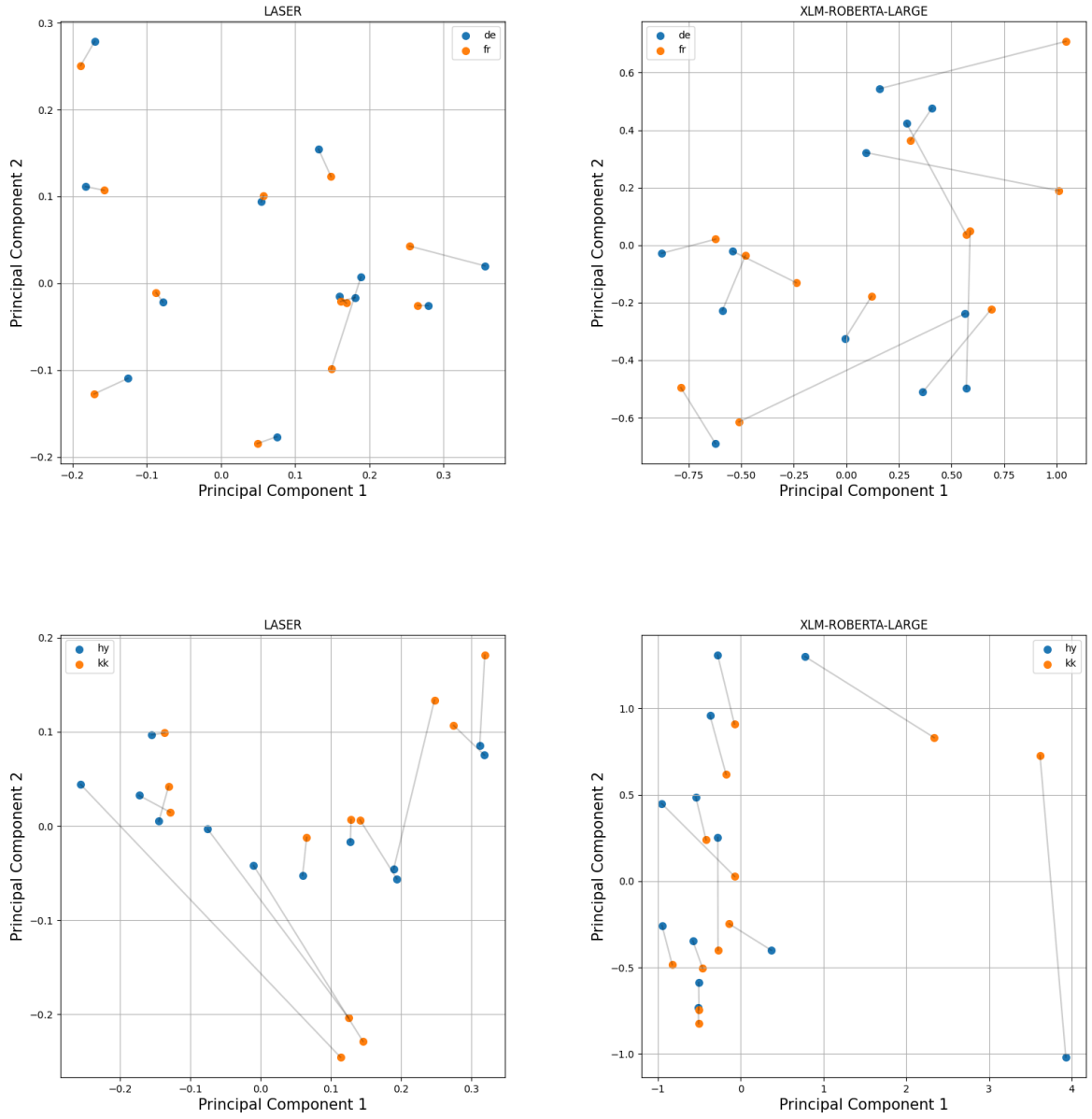

Figure 4: Principal Component Analysis of embeddings extracted from LASER and XLM-R on high-resource (de, fr) and low-resource (hy, kk) languages. Sentences taken for hy and kk are the first 900 sentences from each pair of languages used for scoring F1 and training; 12 sentences used for plotting. For de and fr are taken randomly from huggingfaces “datasets” -> “xnli” -> “train” -> ”premise”. Top left: LASER for de and fr, top right: XLM-R for de and fr, bottom left: LASER for hy and kk, bottom right: XLM-R for hy and kk.

As for Figure 4, it can be seen that LASER for de and fr gives significantly closer embeddings, but it fails on hy and kk by giving embeddings with more considerable distances.

Meanwhile, XLM-R performs almost the same way both for de-fr and hy-kk pairs. However, it provides fewer embeddings with a more considerable distance for hy-kk, compared with LASER.

To conclude the obtained results for PCA pics, we can state that it is visually seen that LASER performs great on high-resource languages and XLM-R for low-resource languages. Still, we do not limit the experiments with only visual representations and calculate Distance Matrices in the coming Section.

3.3.2 Distance Matrices

The Vecalign and the parasite share a common idea: to extract vector representations of each sentence in a parallel document and construct their distance matrix by calculating the distances between each pair(s) and corresponding the pair(s) with the lowest distance to each other. The way of choosing the best matches in both algorithms can be either greedy or by dynamic programming. As has been already mentioned the Vecalign uses LASER as the multilingual pre-trained model for embedding extraction, and the parasite uses XLM-R. The second significant difference is the cost function, which is used to score the distance of a given source and target sentence pair, for which the Vecalign uses the cost function provided in Section 3.2; meanwhile, the parasite records better results using Cosine distance over Euclidean distance. To see how these scoring functions perform in our context, we present distance matrices for the 3 distance measures over 2 language pairs: de-fr and hy-kk.

Figure 5 and Figure 6 show that XLM-R mostly gives homogeneous representations; nevertheless, LASER starts to become biased when producing embeddings for low-resource language pairs. Significantly being biased towards hy, it can be seen from the small rectangle highlighted in the top left of the rectangle, which indicates that hy sentences have a lower distance from themselves compared with kk sentences from themselves.

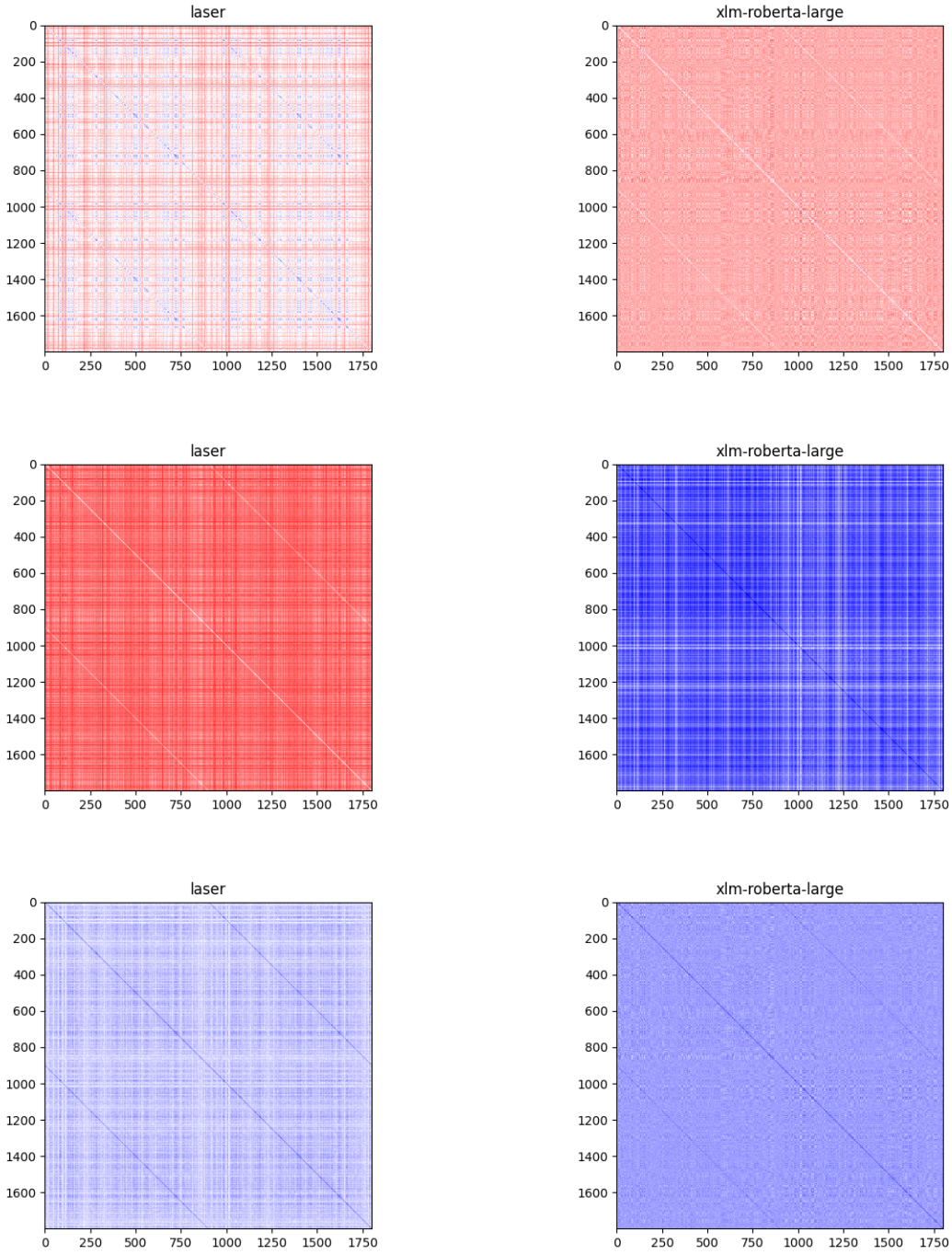

Figure 5: Distance matrix construction, on both horizontal and vertical axes, is a concatenation of source (de) and target (fr) sentence embeddings. From top to bottom, cosine, euclidean, and Vecalign's custom function. From the left are LASER obtained embeddings and from the right XLM-R with center-zero normalization. Note: the colormap used is Matplotlb's "seismic," where blue indicates low value, white medium, and red high.

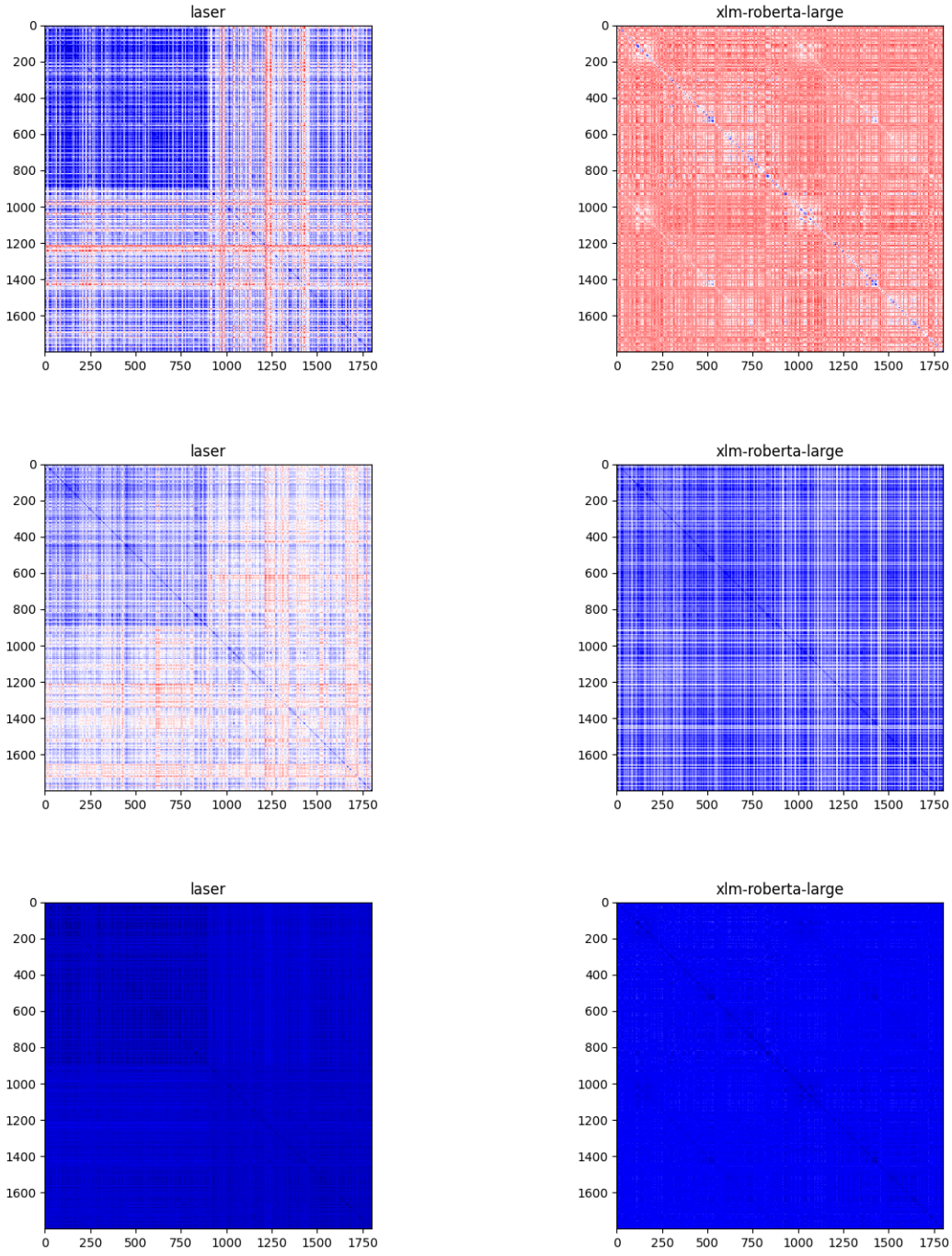

Figure 6: Distance matrix construction, on both horizontal and vertical axes, is a concatenation of source (hy) and target (kk) sentence embeddings, from top to bottom, cosine, euclidean, and Vecalign's custom function. From the left are LASER obtained embeddings and from the right XLM-R with center-zero normalization. Note: the colormap used is Matplotlb's "seismic," where blue indicates low value, white medium, and red high.

The main diagonal, starting from the left top corner and finishing at the bottom right, should always be highlighted and be 0 because that indicates the distance of each sentence from it. The secondary diagonals are the ones indicating the space between two language sentences. One of them starts from the middle of the top edge of the square and ends at the middle of the right edge; this indicates the difference between source and target. The second secondary diagonal starts from the middle of the left edge of the square and finishes at the middle of the square's bottom edge, which shows the difference between target and source pairs. Because the distance operation is commutative, if a secondary diagonal is highlighted, then the second one should also be seen.

Let's look at the Vecalign score function produced distance matrices. We can see that in the case of de-fr using LASER and XLM-R, the secondary diagonals are highlighted. XLM-R slightly faded compared with LASER. Nevertheless, for hy-kk, LASER is left with only one; the main diagonal and XLM-R still have secondary diagonals highlighted.

To give a quantitative representation of these relations, we measure the average similar sentence and dissimilar sentences distances. The similar sentences lay on the secondary diagonal; meanwhile, the dissimilar sentences occupy the area under the secondary diagonal.

It should be noted that the resulting distances are not comparable because of the range they take. That is why they are presented in separate tables. The Cosine distance ranges from 0 to 1, Euclidean from 0 to some maximum possible discrepancy value, and the Vecalign score distance from 0 to n*m (where n and m are the sources and target sentence lengths). We tried to remedy the issue by normalizing the Vecaling.

LASER records higher distances when moving from high-resource to low-resource language pair in the case of Cosine distance. Moreover, LASER is pretty stable for high-resource pairs with significantly lower variance/std. XLM-R, meanwhile, is pretty stable of language pair change. Almost the same picture is for dissimilar distance in XLM-R, plus gives pretty good scores. For high-resource pairs, difference or separability is the same despite the multilingual model. For LASER, the difference decreases by 37% when moving to a low-resource pair. While in the case of XLM-R, the difference is not significant.

For Euclidean distance, LASER decreases score by 20% when moving to a low-resource domain; meanwhile, XLM-R records an increase by 30%; however, the given score for similars distance is lower on the de-fr dataset compared with the hy-kk set. Interestingly, Vecalign score distance LASER and XLM-R do not significantly differ by their difference value for the low-resource language pair, though LASER has significantly higher variance. Nevertheless, again LASER drops its performance when moving to the hy-kk pair by almost 46%.

It is noted that both Cosine and Euclidean distances are stable measures, which is not the case for Vecalign score distance, where a random choice of sentences is used for summations.
The full report of measurements can be found in Appendix Table 1, 2, and 3.

3.4 Alignment Scores

To compare LASER and XLM-R embeddings alignments using the Vecalign algorithm, we picked 3 pairs of languages.

The first one was high-resource languages pair; de and fr, taken from Vecalign's paper, yearbook articles by the Swiss Alpine Club from the Text+Berg corpus. We obtained the identical scores reported in Vecaling for LASER embeddings alignment for de→fr and significantly lower score when using XLM-R.

The next pair was one high-resource (en) and one low-resource (kk) languages; we used WMT19 new translation datasets provided by the organizers. We found the source of the provided dataset and diversified the documents used in the test set from validation and training sets. Made alignments from LASER and XLM-R extracted embeddings and compared them with the given ideal test set. The set provided by the organizers was ideally clean and aligned. We eventually obtained almost identical scores, with a slight advantage to XLM-R.

The last pair contained low-resource languages; hy and kk. The findings of parallel data in these two languages were not an easy one. The one found was a report from the "Eurasian Economic Union," which was supposed to be clean and almost identical in all languages because of being an official document. The kk version of that document we found from a non-official source had some extra sentences (links to external articles). Despite this, there were alignment problems, so though it was an official document, there were still differences, so it was a perfect dataset to do experiments for this problem. The validation set is hand-aligned, and the ladder file is manually created. The obtained results were remarkable; XLM-R exceeded LASER by almost the same difference for de→fr. Table 6 presents the results of precision, recall, and F_1 score for 3 pairs.

	de→fr			en→kk			hy→kk		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
LASER	0.90	0.90	**0.90**	0.84	0.73	0.78	0.80	0.78	0.79
XLM-R	0.75	0.79	0.76	0.84	0.74	**0.79**	0.98	0.99	**0.98**

Table 6: Precision, Recall, and F_1 scores on de→fr (high-resource), en→kk (high and low-resources), and hy→kk (low-resource) pairs corpus alignment, embeddings extracted from LASER and XLM-R models, respectively.

F_1 score is known from statistical analysis of binary classification. It is obtained from the precision and recall of the test. Precision intuitively can be defined as the number of selected items relevant, and recall is the number of relevant items selected. F_1 varies between 0 and 1; the larger, the better. So F_1 is given by the expression:

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}}$$

Alignments in Vecalign are outputted in ladder-like format, and scoring is also done by accepting gold/ideal alignment ladder-style file. To see what a ladder-style file looks like, let us exhibit the parallel document in Table 1. After segmentation, the first sentence in English (the title) will remain steady, but the word “OBJECTIVE” will appear in the second line; the remaining will be the same as it appears in Table 1. So the ladder-style file for ideal alignment for this document will be the following:

```
[0]:[]
[1, 2]:[0]
[3]:[1, 2]
[4]:[3]
[5]:[4]
```

Table 7: Ladder-style file of Table 1 document.

In square brackets, the indexes of sentences are given; if there are more than two will be separated by a comma and space. A block (the object surrounded by square brackets) corresponds to another block, and they are divided by a colon from each other. If a block of sentences does not match any block, the corresponding block should be empty.

We have also experimented with different variations of sentence embedding extraction from tokens’ embeddings, such as using mean-pooling, max-pooling, using MLM head as the

representation of the sentence, adding an extra mask token, center zeroing/normalizing, centering one sentence from another, etc. Based on those experiments, we continually used XLM-R with mean pooling and center zeroing as the best performing method for sentence embedding extraction. The full table of experiments can be found in Appendix Table 4.

3.5 Downstream Task

To show the effect of parallel data quality and the difference between LASER and XLM-R on low-resource languages, we demonstrate a downstream task; NMT for en↔kk pair.
The set used is taken from the WMT19 low-resource English↔Kazakh translation task. Organizers provide both the training and validation sets. The datasets have around 7700 sentences in the training set and 2100 in the validation set.

The Architecture used for this task is “transformer_wmt_en_de_big” with default settings that we used for WMT20 training, using ADAM optimizer, without label smoothing, with 10^{-4} learning rate inverse square root learning rate scheduler. With the same approach for our WMT20 submissions, the training was terminated if the validation loss did not improve for 5 consecutive epochs. The results are presented in Table 8.

It is worth mentioning that the choice of the dataset and the model is not the perfect one. Because the dataset does not contain “sufficiently” many samples and the architecture is immense, having 213M parameters makes it easy to overfit the model and/or not get a good performance.

	en→kk	**kk→en**
Original	0.8	2.2
Vecalign aligned (LASER)	0.9	**2.4**
Vecalign aligned (XLM-R)	**1.1**	2.3

Table 8: Results of BLEU scores on official WMT19 en↔kk test set for our system. The original presents the original data provided by the organizers (all the preprocessing steps are applied except alignment), with no alignment applied on our side. The following records are the aligned results using LASER and XLM-R embeddings.

Results show that Vecalign improves the BLEU score in both directions, whether using LASER or XLM-R. The XLM-R pioneered in the en→kk direction; meanwhile, LASER scored a slightly better BLEU score for the kk→en direction.

Compared with the previous year's WMT participating model ("The NiuTrans Machine Translation Systems for WMT19" Li et al., 2019), our systems recorded significantly lower BLEU scores; 2.6 for en→kk direction and 10.1 for kk→en.

We explain this by the architectural difference. The NEU team's architecture outperforms the standard Transformer-Big significantly in translation quality and convergence speed (Li et al., 2019). This architecture is a modification of the Transformer-Base, which has around 65M parameters (Wang et al., 2019). Because the main purpose of this experiment was not to outperform the NEU team's results but to show the effect of alignment, we didn't continue training with their specified architecture.

Conclusion

This work represents the design of machine translation systems during the Summer internship at YerevaNN and the more in-depth research on bilingual sentence alignment.

We described our submission for the WMT20 Biomedical Translation task, which includes a new algorithm, "parasite," for bilingual sentence alignment based on multilingual language models. According to the human evaluation, we have recorded the best BLEU scores in English↔Russian directions according to the automatic evaluation and had strong results in English↔Russian and English↔German directions. These results were made possible because of the parallel sentence alignment and the heavy preprocessing pipeline we used.

We investigated commonly used bilingual sentence alignment algorithms, presenting their differences, both ideologically and computationally. We applied Vecalign, a recently proposed alignment algorithm, on German↔French, English↔Kazakh, and Armenian↔Kazakh language pairs and measured alignment accuracy while using LASER and XLM-R pre-trained embeddings. We showed the advantage of XLM-R extracted embeddings for a low-resource language pair over LASER. The advantage was further confirmed by PCA visualizations and pairwise distance matrices of the sentence embeddings in various languages. Finally, we trained machine translation systems using the obtained alignments to explore the effect of the alignments on a downstream task.

The experiments performed in this work showed that in terms of bilingual sentence alignment, LASER is getting worse on low-resource languages and low-resource (e.g., biomedical) domains of rich languages; meanwhile, XLM-R's performance is relatively constant.

Most of the scripts written for these experiments can be found on:
https://github.com/HovhannesTamoyan/billingual_sentence_alignment,
https://github.com/HovhannesTamoyan/wmt20_biomedical_scripts,
https://github.com/YerevaNN/parasite.

Acknowledgments

We would like to thank Hrant Khachatryan from YerevaNN for valuable discussions on the bilingual sentence alignment problem and Adam Bittlingmayer from ModelFront for helpful discussions on the quality of parallel corpora. We are also grateful for the 2x Titan V GPUs donated to YerevaNN by NVIDIA we used for all the experiments.

References

Artetxe, M. and Schwenk, H., 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. Transactions of the Association for Computational Linguistics, 7, pp.597-610.

Barrault, L., Bojar, O., Costa-Jussa, M.R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S. and Monz, C., 2019, August. Findings of the 2019 conference on machine translation (wmt19). In Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1) (pp. 1-61).

Bawden, R., Cohen, K.B., Grozea, C., Yepes, A.J., Kittner, M., Krallinger, M., Mah, N., Neveol, A., Neves, M., Soares, F. and Siu, A., 2019, August. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)* (pp. 29-53).

Bawden, R., Di Nunzio, G., Grozea, C., Unanue, I., Yepes, A., Mah, N., Martinez, D., Névéol, A., Neves, M., Oronoz, M. and de Viñaspre, O.P., 2020. Findings of the WMT 2020 Biomedical Translation Shared Task: Basque, Italian and Russian as New Additional Languages. In 5th Conference on Machine Translation.

Buck, C. and Koehn, P., 2016, August. Findings of the WMT 2016 bilingual document alignment shared task. In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers (pp. 554-563).

Braune, F. and Fraser, A., 2010, August. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In Coling 2010: Posters (pp. 81-89).

Brown, P.F., Lai, J.C. and Mercer, R.L., 1991, June. Aligning sentences in parallel corpora. In 29th Annual Meeting of the Association for Computational Linguistics (pp. 169-176).

Cer, D., Yang, Y., Kong, S.Y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C. and Sung, Y.H., 2018. Universal sentence encoder. arXiv preprint arXiv:1803.11175.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L. and Stoyanov, V., 2019. Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116.

Edunov, S., Ott, M., Auli, M. and Grangier, D., 2018. Understanding back-translation at scale. arXiv preprint arXiv:1808.09381.

Gale, W.A. and Church, K., 1991. Identifying word correspondences in parallel texts. In Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991.

Gomes, L. and Lopes, G., 2016, May. First steps towards coverage-based sentence alignment. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) (pp. 2228-2231).

Guo, M., Shen, Q., Yang, Y., Ge, H., Cer, D., Abrego, G.H., Stevens, K., Constant, N., Sung, Y.H., Strope, B. and Kurzweil, R., 2018. Effective parallel corpus mining using bilingual sentence embeddings. arXiv preprint arXiv:1807.11906.

Hambardzumyan, K., Tamoyan, H. and Khachatrian, H., 2020, November. YerevaNN's Systems for WMT20 Biomedical Translation Task: The Effect of Fixing Misaligned Sentence Pairs. In *Proceedings of the Fifth Conference on Machine Translation* (pp. 820-825).

Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M. and Liu, S., 2018. Achieving human parity on automatic chinese to english news translation. arXiv preprint arXiv:1803.05567.

Khayrallah, H. and Koehn, P., 2018. On the impact of various types of noise on neural machine translation. arXiv preprint arXiv:1805.12282.

Kudo, T. and Richardson, J., 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226.

Kvapilíková, I., Artetxe, M., Labaka, G., Agirre, E. and Bojar, O., 2020, July. Unsupervised Multilingual Sentence Embeddings for Parallel Corpus Mining. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop (pp. 255-262).

Lazaridou, A., Chrupała, G., Fernández, R. and Baroni, M., 2016. Multimodal semantic learning from child-directed input. In Knight K, Nenkova A, Rambow O, editors. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2016 Jun 12-17; San Diegio, California. Stroudsburg (PA): Association for Computational Linguistics; 2016. p. 387–92. ACL (Association for Computational Linguistics).

Li, B., Li, Y., Xu, C., Lin, Y., Liu, J., Liu, H., Wang, Z., Zhang, Y., Xu, N., Wang, Z. and Feng, K., 2019, August. The NiuTrans machine translation systems for WMT19. In Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1) (pp. 257-266).

Melamed, I.D., 1996. A geometric approach to mapping bitext correspondence. arXiv preprint cmp-lg/9609009.

Müller, R., Kornblith, S. and Hinton, G., 2019. When does label smoothing help?. arXiv preprint arXiv:1906.02629.

Neumann, M., King, D., Beltagy, I. and Ammar, W., 2019. Scispacy: Fast and robust models for biomedical natural language processing. arXiv preprint arXiv:1902.07669.

Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M. and Edunov, S., 2019. Facebook FAIR's WMT19 News Translation Task Submission. arXiv preprint arXiv:1907.06616.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D. and Auli, M., 2019. fairseq: A fast, extensible toolkit for sequence modeling. arXiv preprint arXiv:1904.01038.

Radovanovic, M., Nanopoulos, A. and Ivanovic, M., 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research, 11(sept), pp.2487-2531.

Reimers, N. and Gurevych, I., 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Saunders, D., Stahlberg, F. and Byrne, B., 2019. UCAM Biomedical translation at WMT19: Transfer learning multi-domain ensembles. arXiv preprint arXiv:1906.05786.

Sennrich, R. and Volk, M., 2010. MT-based sentence alignment for OCR-generated parallel texts.

Simard, M. and Plamondon, P., 1998. Bilingual sentence alignment: Balancing robustness and accuracy. Machine Translation, 13(1), pp.59-80.

Thompson, B. and Koehn, P., 2019, November. 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) (pp. 1342-1348).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. Attention is all you need. arXiv preprint arXiv:1706.03762.

Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L. and Trón, V., 2007. Parallel corpora for medium density languages. Amsterdam Studies In The Theory And History Of Linguistic Science Series 4, 292, p.247.

Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D.F. and Chao, L.S., 2019. Learning deep transformer models for machine translation. arXiv preprint arXiv:1906.01787.

Zweigenbaum, P., Sharoff, S. and Rapp, R., 2018, May. Overview of the third BUCC shared task: Spotting parallel sentences in comparable corpora. In Proceedings of 11th Workshop on Building and Using Comparable Corpora (pp. 39-42).

Appendix

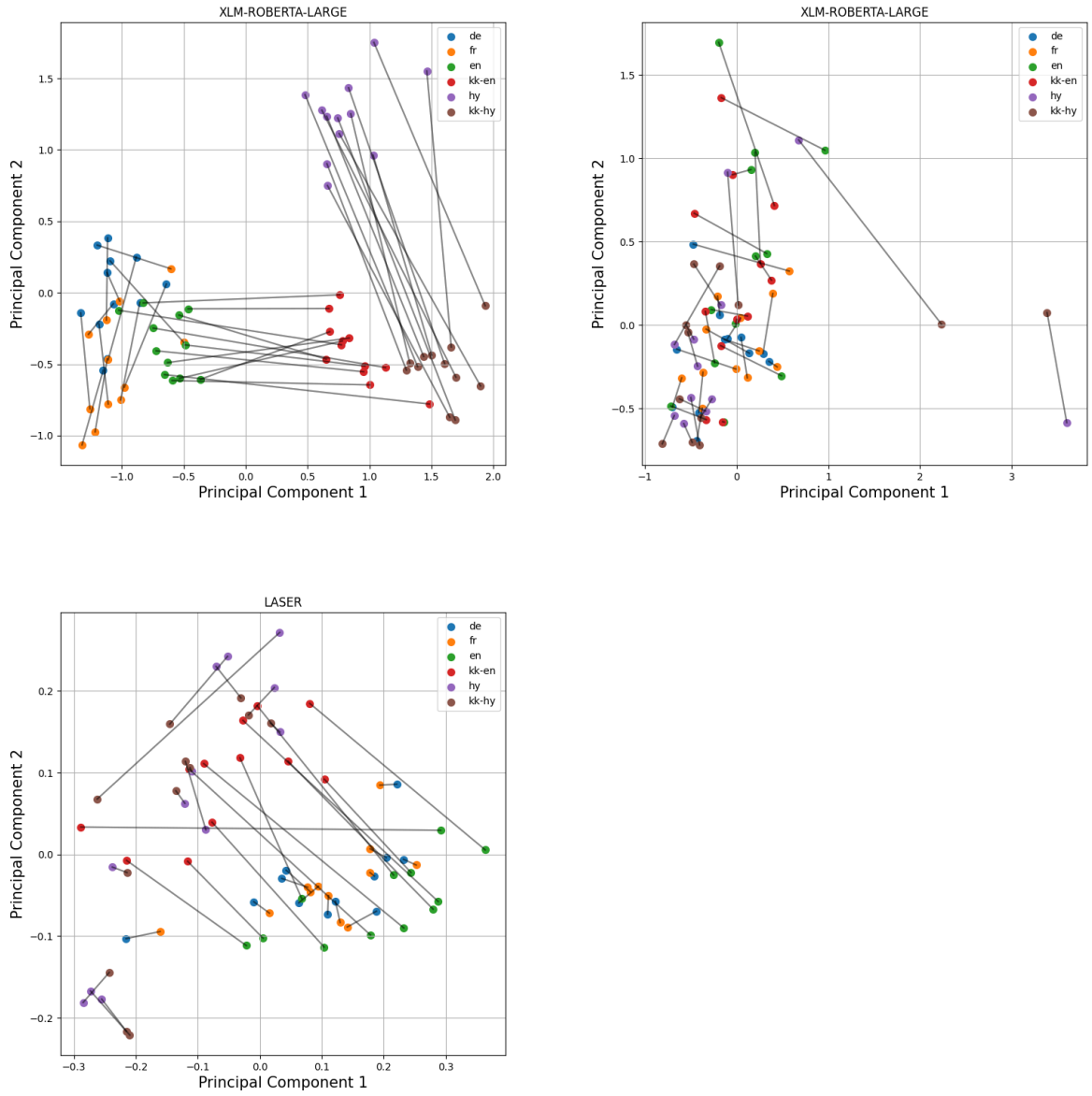

Figure 1: Principal Component Analysis of embeddings extracted from LASER and XLM-R on 5 languages (en, de, fr, kk, hy). Sentences taken for en, kk, and hy are the first 900 sentences from each pair of languages used for scoring F1 and training; 12 sentences used for plotting. For de and fr are taken randomly from huggingfaces “datasets” -> “xnli” -> “train” -> ”premise”. Because there was no pre-trained Sentence Transformer found trained on the low resources (kk, hy), it was impossible to provide their visualization. Top left: XLM-R with mean pooling, top right: XLM-R with mean pooling and embeddings centered zero, bottom left: LASER. Note: kk-en is the kk from en-kk set and kk-hy is kk from hy-kk set.

		Similars distance	Dissimilar distance	Difference
de-fr	LASER	0.09 ± (0.059)	0.52 ± (0.272)	0.43
	XLM-R	0.56 ± (0.155)	0.99 ± (0.511)	0.43
hy-kk	LASER	0.28 ± (0.150)	0.44 ± (0.263)	0.16
	XLM-R	0.58 ± (0.158)	0.97 ± (0.506)	0.39

Table 1: The Cosine distance between source and target languages, using LASER and XLM-R extracted embeddings. The similars' distance shows the semantically close, but in different languages, sentences mean. The dissimilar distance shows the semantically not close and in different languages sentences mean. On the braces, the std of each list is given. The difference is the subtraction of dissimilar distances from a similar distance. Note: the lower the similarities distance and the higher the dissimilar distance, the better. The higher the difference between these two measures, the better.

		Similars distance	Dissimilar distance	Difference
de-fr	LASER	0.24 ± (0.063)	0.64 ± (0.325)	0.39
	XLM-R	1.73 ± (0.629)	2.31 ± (1.235)	0.58
hy-kk	LASER	0.32 ± (0.076)	0.41 ± (0.218)	0.08
	XLM-R	2.26 ± (1.013)	3.08 ± (1.779)	0.82

Table 2: The Euclidean distance between source and target languages, using LASER and XLM-R extracted embeddings. The similars' distance shows the semantically close, but in different languages, sentences mean. The dissimilar distance shows the semantically not close and in different languages sentences mean. On the braces, the std of each list is given. The difference is the subtraction of dissimilar distances from a similar distance. Note: the lower the similarities distance and the higher the dissimilar distance, the better. The higher the difference between these two measures, the better.

		Similars distance	Dissimilar distance	Difference
de-fr	LASER	0.02 ± (0.017)	0.16 ± (0.085)	0.13
	XLM-R	0.09 ± (0.027)	0.16 ± (0.086)	0.07
hy-kk	LASER	0.12 ± (0.055)	0.18 ± (0.104)	0.06
	XLM-R	0.10 ± (0.029)	0.16 ± (0.088)	0.06

Table 3: The Vecalign score distance between source and target languages, using LASER and XLM-R extracted embeddings. The similars' distance shows the semantically close, but in different languages, sentences mean. The dissimilar distance shows the semantically not close and in different languages sentences mean. On the braces, the std of each list is given. The difference is the subtraction of dissimilar distances from a similar distance. Note: the lower the similarities distance and the higher the dissimilar distance, the better. The higher the difference between these two measures, the better.

Name	pooling	mask_token_added	use_encoder	use_mlm_head	center_zero	P	R	F_1
avg.without_encoding	avg					0.12	0.11	0.115
max.without_encoding	max					0.40	0.44	0.422
center_zero.avg.without_encoding	avg				y	0.13	0.14	0.137
center_zero.max.without_encoding	max				y	0.05	0.08	0.063
avg	avg		y			0.65	0.70	0.669
max	max		y			0.54	0.61	0.573
mask	mask	y	y			0.66	0.70	0.677
center_zero.avg	avg		y		y	**0.75**	**0.79**	**0.769**
center_zero.max	max		y		y	0.18	0.26	0.210
center_zero.mask	mask	y	y		y	**0.75**	0.77	0.761
mask.use_mlm_head	mask	y	y	y		0.62	0.66	0.640
center_zero.mask.use_mlm_head	mask	y	y	y	y	0.73	0.74	0.735
center_zero_by_lang.mask.use_mlm_head	mask	y	y	y	by lang	0.51	0.58	0.542

Table 4: The sentence embedding construction from XLM-R given token embeddings. Note: y indicates yes.