PMI-Align: Word Alignment With Point-Wise Mutual Information Without Requiring Parallel Training Data

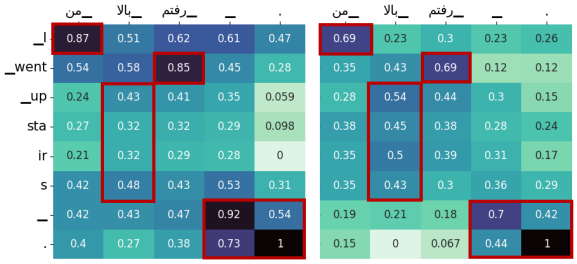
Fatemeh Azadi, Heshaam Faili, and Mohammad Javad Dousti
School of Electrical and Computer Engineering, University of Tehran, Iran
{ft.azadi, hfaili, mjdousti}@ut.ac.ir

Abstract

Word alignment has many applications including cross-lingual annotation projection, bilingual lexicon extraction, and the evaluation or analysis of translation outputs. Recent studies show that using contextualized embeddings from pre-trained multilingual language models could give us high quality word alignments without the need of parallel training data. In this work, we propose PMI-Align which computes and uses the point-wise mutual information between source and target tokens to extract word alignments, instead of the cosine similarity or dot product which is mostly used in recent approaches. Our experiments show that our proposed PMI-Align approach could outperform the rival methods on five out of six language pairs. Although our approach requires no parallel training data, we show that this method could also benefit the approaches using parallel data to fine-tune pre-trained language models on word alignments. Our code and data are publicly available[1].

1 Introduction

Word alignment, as the task of finding the corresponding source and target tokens in a parallel sentence, was well-known as an essential component of statistical machine translation (SMT) systems. Despite the dominance of neural machine translation (NMT) in recent years, word alignment is still a notable area of research due to its usage in a wide variety of NLP applications, such as annotation projection (Yarowsky et al., 2001; Padó and Lapata, 2009; Huck et al., 2019; Nicolai and Yarowsky, 2019), bilingual lexicon extraction (Ammar et al., 2016; Shi et al., 2021; Artetxe et al., 2019), typological analysis (Lewis and Xia, 2008; Östling, 2015), guided alignment training of NMT (Liu et al., 2016; Chen et al., 2016; Alkhouli et al., 2018), and evaluation and analysis of translation outputs (Anthony et al., 2019; Neubig et al., 2019; Wang et al., 2020).



(a) (b)

Figure 1: Similarity matrix consists of cosine similarities between subword representations (a) vs. PMI matrix (b) for an English-Persian sentence pair. Both measures are normalized with min-max normalization to be comparable. Red boxes denote the gold alignments.

For many years statistical methods such as IBM models (Brown et al., 1993) and tools implemented based on them, namely GIZA++ (Och and Ney, 2003) or fast-align (Dyer et al., 2013), were among the most popular solutions to the word alignment task. Following the rise of deep neural models, several attempts have been made to extract word alignments from NMT models and their attention matrices (Peter et al., 2017; Ghader and Monz, 2017; Zenkel et al., 2020; Zhang and van Genabith, 2021). However, most of these methods, as well as the statistical aligners, require a sufficient amount of parallel training data to produce high quality word alignments. Recently, Jalili Sabet et al. (2020) have shown that high quality word alignments could be achieved using pre-trained multilingual language models (LMs), like MBERT Devlin et al. (2019) and XLMR Conneau et al. (2020). Their proposed method called SimAlign, extracts word alignments from similarity matrices induced from multilingual contextualized word embeddings with no need for parallel training data, which is very useful for low-resource language pairs. Afterwards, Dou and Neubig (2021) and Chi et al. (2021) proposed methods

[1] https://github.com/fatemeh-azadi/PMI-Align

Findings of the Association for Computational Linguistics: ACL 2023, pages 12366–12377

called probability thresholding and optimal transport to extract alignments using the similarity matrices derived from pre-trained LMs. They have also proposed some word alignment objectives to fine-tune the pre-trained models over parallel corpora.

In this paper, we follow the work done by Jalili Sabet et al. (2020) to extract alignments from pre-trained LMs without requiring any parallel training data and propose *PMI-Align*. Our main contribution is proposing to compute the *point-wise mutual information* (PMI) between source and target tokens and using the PMI matrices instead of similarity matrices made of cosine similarities between the representation vectors of each source and target tokens, to align words. We argue that our proposed PMI-based method could align better as it considers the total alignment probability of each source or target token, as well as the joint alignment probabilities (equivalent to cosine similarities). This could alleviate the so-called hubness problem (Radovanovic et al., 2010) in high dimensional spaces, where some token's representation is close to many others (see *_went* in Figure 1). We perform experiments on six different language pairs and show that our method could surpass other alignment methods on five of them. We also conduct our experiments on different pre-trained LMs to show that PMI-Align could be advantageous regardless of the pre-trained model used.

2 Proposed Method

In this section, we first discuss how we define and compute the PMI matrix for each sentence pair and then we describe our alignment extraction method using the PMI matrix.

2.1 Point-Wise Mutual Information

Point-wise mutual information (PMI) is a well-known measure of association in information theory and NLP and it shows the probability of two events x and y occurring together, compared to what this probability would be if they were independent (Fano, 1961). It is computed as follows:

$$PMI(x,y) := \log \frac{p(x,y)}{p(x)p(y)} \tag{1}$$

In the context of word alignments, we define the PMI for a source and target token in a sentence pair as how more probable two tokens are to be aligned than if they are aligned randomly. Given a sentence $x =< x_1,...,x_n >$ in the source language and its corresponding target sentence $y =< y_1,...,y_m >$, the joint alignment probability of two tokens, x_i and y_j, could be computed as:

$$p(x_i,y_i) = \frac{e^{sim(h_{x_i},h_{y_j})}}{\sum_{i',j'} e^{sim(h_{x_{i'}},h_{y_{j'}})}}, \tag{2}$$

where h_{x_i} is the contextualized embedding vector of x_i extracted from a pre-trained multilingual language model and $sim(.)$ is the cosine similarity measure. The total alignment probability of x_i and y_j, i.e., $p(x_i)$ and $p(y_j)$, could also be computed according to the total probability rule as follows:

$$p(x_i) = \sum_{1\leq j\leq m} p(x_i,y_j) \tag{3}$$

By calculating the PMI for each source and target token in a parallel sentence, we obtain the PMI matrix for that sentence pair, that could be used to extract alignments instead of similarity matrix in SimAlign (Jalili Sabet et al., 2020). The advantage of using PMI to align words is that it also considers the total alignment probability of each source and target token in addition to their joint alignment probability, which is equivalent to the similarity measure. This leads to reduce the probability to align the token pairs that one of them has high similarities to many other tokens, and thus could alleviate the so-called hubness problem in high dimensional spaces where some data points called hubs are the nearest neighbors of many others.

2.2 Extracting Alignments

To extract word alignments, we follow the simple Argmax method proposed in Jalili Sabet et al. (2020). Thus, we first obtain the source to target and target to source alignment matrices using the argmax over each row and each column of the PMI matrix, respectively. Next, we intersect these two matrices to get the final word alignment matrix. In other words, the final alignment matrix $A_{ij} = 1$ iff $i = argmax_k(PMI_{kj})$ and $j = argmax_k(PMI_{ik})$.

Since the above method would extract alignments on the subword level, we follow the heuristic used in previous work to obtain the word-level alignments by considering two words to be aligned if any of their subwords are aligned (Jalili Sabet et al., 2020; Zenkel et al., 2020; Dou and Neubig, 2021).

3 Experiments and Results

3.1 Datasets

We perform our experiments on six public datasets, as in (Jalili Sabet et al., 2020), consists of English-Czech (En-Cs), German-English (De-En), English-Persian (En-Fa), English-French (En-Fr), English-Hindi (En-Hi) and Romanian-English (Ro-En) language pairs. The statistics and URLs of these datasets are available in Table 2 in Appendix A.

3.2 Models and baselines

We compare our method with the following three state-of-the-art methods proposed to extract alignments from pre-trained multilingual LMs without using parallel training data. For all these methods default parameters were used in our experiments.

SimAlign[2] (Jalili Sabet et al., 2020): They propose three methods to extract alignments from similarity matrices, called Argmax, Itermax and Match. Although Itermax and Match methods could not make significant improvements over Argmax and the Argmax method had better AER results for most of language pairs while using the XLMR-base model, they have argued that the Itermax method, which tries to apply Argmax iteratively, could be beneficial for more distant language pairs. Thus, we report both Argmax and Itermax results in our experiments to compare with our method.

Probability Thresholding[3] (Dou and Neubig, 2021): In this method they apply a normalization function, i.e., softmax, to convert the similarity matrix of tokens into source to target and target to source alignment probability matrices. Afterwards, they extract the aligned words as the words that their alignment probabilities in both matrices exceed a particular threshold.

Optimal Transport[4] (Chi et al., 2021): This method was proposed in both Dou and Neubig (2021) and Chi et al. (2021), and tried to model the word alignment task as the known optimal transport problem (Cuturi, 2013). Using the similarity matrix, this method attempted to find the alignment probability matrix that maximizes the sentence pair similarity. In our experiments, we use the method proposed by Chi et al. (2021) that utilizes the regularized variant of the optimal transport problem (Peyré et al., 2019), as it reported better results.

There are also many attempts made to improve the pre-trained LMs by fine-tuning on some parallel corpora to better align words. However, as our approach is irrelevant to the pre-trained model and our focus is on the alignment extraction instead of the model, we do not include those methods in our experiments. To demonstrate the effectiveness of our PMI-based alignment regardless of the utilized pre-trained multilingual LM, we conduct our experiments on M-BERT (Devlin et al., 2019), XLMR-Base (Conneau et al., 2020) and XLM-Align (Chi et al., 2021) which is fine-tuned on a word-alignment task, to show that our method could also be advantageous on more cross-lingually aligned models. All these models are publicly available in the Hugging Face platform (Wolf et al., 2020).

3.3 Results

Table 1 shows the results of our alignment technique compared to previous methods while using different pre-trained LMs. Following the previous work (Jalili Sabet et al., 2020; Dou and Neubig, 2021; Chi et al., 2021), we use the 8th layer's representations of each pre-trained model to compute the similarity or PMI matrices. We also use the alignment error rate (AER) (Och and Ney, 2003) as the evaluation metric.

As Table 1 shows, our PMI-Align method could consistently outperform the other methods in all language pairs except En-Fr, regardless of the pre-trained model used. Compared to Argmax, our method performs better for about 1% or more in AER, while using the XLMR-Base model (except for En-Fr), which exclusively shows the benefits of using the PMI matrix instead of the similarity matrix. We also see that the PMI-Align could surpass the Itermax method for more distant language pairs such as En-Fa and En-Hi, where it was claimed to have the most advantage. Results show that our method could also be beneficial while using a model pre-trained on a word alignment task, i.e., XLM-align, which is expected to have more cross-lingually aligned representations, and less hubness problem.

The only language pair that our method could

[2] https://github.com/cisnlp/simalign
[3] https://github.com/neulab/awesome-align
[4] https://github.com/CZWin32768/XLM-Align

Pretrained Model	Alignment method	Aignment Error Rate						Avg
		En-Cs	De-En	En-Fa	En-Fr	En-Hi	Ro-En	
M-BERT	SimAlign - Argmax	12.8	18.5	37.1	5.8	44.1	34.4	25.5
	SimAlign - Itermax	15.0	19.0	33.8	9.0	41.3	31.2	24.9
	Probability Thresholding	12.6	17.4	33.9	**5.6**	41.2	32.1	23.8
	Optimal Transport	12.9	17.8	33.9	6.0	40.9	31.7	23.9
	PMI-Align	11.8	17.0	32.8	5.7	39.3	30.9	22.9
XLMR-Base	SimAlign - Argmax	12.5	18.9	30.2	6.4	38.8	28.2	22.5
	SimAlign - Itermax	15.0	20.2	29.1	10.0	38.7	27.4	23.4
	Probability Thresholding	17.4	23.1	35.0	9.2	42.6	32.0	26.6
	Optimal Transport	12.3	17.7	29.0	7.5	37.9	27.5	22.0
	PMI-Align	11.7	17.4	28.1	7.3	37.5	26.8	21.5
XLM-Align	SimAlign - Argmax	10.7	16.6	28.4	**5.6**	34.6	27.7	20.6
	SimAlign - Itermax	14.1	18.9	27.6	10.3	33.8	27.1	22.0
	Probability Thresholding	13.7	18.5	29.6	7.9	35.2	28.4	22.2
	Optimal Transport	11.1	16.6	28.0	6.6	34.0	27.0	20.6
	PMI-Align	**10.4**	**16.0**	**26.7**	6.2	**33.4**	**26.3**	**19.8**

Table 1: AER results of our PMI-Align method compared to the other alignment extraction methods on 6 language pairs, while using different pre-trained models. The overall best results are in bold.

not outperform prior methods is En-Fr. This could be due to the closeness of these two languages, as they have many shared subwords and similar word orderings. As a result, pre-trained models for this language pair are better trained and could strongly produce similar representations for aligned words, which reduces the hubness problem to a great extent. Thus, using PMI instead of the similarity matrix could not help. However, our method's performance while using the M-BERT model is comparable to the best results, with about 0.1% difference in AER. Several samples are shown in Appendix B, to better intuitively compare PMI-Align and Argmax, which could better show the benefits of using the PMI matrix instead of the cosine similarities.

4 Related Work

Statistical aligners based on IBM models (Brown et al., 1993), such as Giza++ (Och and Ney, 2003) and fast align (Dyer et al., 2013) were the most dominant tools for word alignment until the late 2010s. With the rise of neural machine translation models, several attempts made to extract alignments from them (Ghader and Monz, 2017; Garg et al., 2019; Li et al., 2019; Zenkel et al., 2020; Chen et al., 2021; Zhang and van Genabith, 2021). However, all these models need parallel training data and could not utilize pre-trained contextualized embeddings. Recently, Jalili Sabet et al. (2020) have proposed methods to extract alignments from similarity matrices induced from multilingual LMs without the need for training on parallel data. Following this work, we propose a PMI measure to score and align words in each sentence pair, instead of cosine similarity. Some other alignment extraction methods using multilingual LMs were also provided by Dou and Neubig (2021) and Chi et al. (2021). They both also proposed several training objectives related to word alignments to fine-tune multilingual LMs on parallel data, as in some other recent works (Cao et al., 2020; Wu and Dredze, 2020; Lai et al., 2022).

5 Conclusions

This paper presents a word alignment extraction method based on the PMI matrices derived from cross-lingual contextualized embeddings, instead of just the similarity matrices. We proposed a way to compute the PMI matrix for each sentence pair and argued that using this PMI measure would be beneficial since for each source-target word pair, it considers not only their similarity to each other but also their similarity values to the other tokens of the sentence, that could mitigate the hubness problem.

Experimental results show that our PMI-Align method could outperform the previous alignment extraction methods in five out of six language pairs, regardless of the base pre-trained language model used to derive word embeddings. Although our method does not require any parallel training data, our experiments show that it could also benefit the approaches using such data to fine-tune the pre-trained models for better word alignments. In future work, the proposed PMI matrix could be investigated in other cross-lingual or even monolingual applications, like the translation quality estimation or the evaluation of text generation tasks, instead of the similarity matrix.

Limitations

Although our proposed aligner has surpassed the existing LM-based alignment extraction methods in most of the datasets, it could not make any improvement for the En-Fr language pair, as shown in Table 1. This suggests that our proposed method might be only beneficial for more distant languages. On the other hand, for similar languages, it not only cannot add any information to the similarity matrix, but also its estimation for the alignment probabilities might add noise to the alignment extraction method. Thus, investigating ways to more effectively estimate the alignment probabilities of source and target tokens might be helpful in future work.

Another limitation of our method, as well as other LM-based aligners, is that they first extract subword-level alignments, and then heuristically map them to word-level. By observing the aligner outputs, we realize that many errors occur when the pre-trained LM can not efficiently split words into meaningful subwords. This happens more often for low-resource languages or far languages from English (like Persian or Hindi). Thus, achieving better subword tokenization in pre-trained LMs or applicable methods to convert subword-level representations into word-level could help improve the quality of LM-based aligners.

References

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 177–185.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*.

Bau Anthony, Belinkov Yonatan, Sajjad Hassan, Durrani Nadir, Dalvi Fahim, Glass James, et al. 2019. Identifying and controlling important neurons in neural machine translation. In *7th International Conference on Learning Representations*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. Bilingual lexicon induction through unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5002–5007.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):261–311.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *ICLR*.

Chi Chen, Maosong Sun, and Yang Liu. 2021. Mask-align: Self-supervised neural word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4781–4791, Online. Association for Computational Linguistics.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *AMTA 2016*.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, He-Yan Huang, and Furu Wei. 2021. Improving pretrained cross-lingual language models via self-labeled word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3418–3430.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Robert M Fano. 1961. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29(11):793–794.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39.

Matthias Huck, Diana Dutka, and Alexander Fraser. 2019. Cross-lingual annotation projection is effective for neural part-of-speech tagging. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 223–233.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643.

Siyu Lai, Zhen Yang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2022. Cross-align: Modeling deep cross-lingual interactions for word alignment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

William Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102.

David Mareček. 2008. Automatic alignment of tectogrammatical trees from czech-english parallel corpus. Master's thesis, Charles University, MFF UK.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond*, pages 1–10.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41.

Garrett Nicolai and David Yarowsky. 2019. Learning morphosyntactic analyzers from the bible via iterative annotation projection across 26 languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1765–1774.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 440–447.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Robert Östling. 2015. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211.

Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Jan-Thorsten Peter, Arne Nix, and Hermann Ney. 2017. Generating alignments using target foresight in attention-based neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108:27–36.

Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531.

Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. 2021. Bilingual lexicon induction via unsupervised bitext construction and word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 813–826.

Leila Tavakoli and Heshaam Faili. 2014. Phrase alignments in parallel corpus using bootstrapping approach. *International Journal of Information and Communication Technology Research*, 6(3).

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3070–3079.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Shijie Wu and Mark Dredze. 2020. Do explicit alignments robustly improve multilingual encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4471–4482.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–8.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms giza++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617.

Jingyi Zhang and Josef van Genabith. 2021. A bidirectional transformer based alignment model for unsupervised word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 283–292.

A Data Statistics

Table 2 shows the number of sentences and the download links of the test datasets we used in our experiments.

B Alignment Examples

Figures 2 and 3 illustrate some sentence pair examples comparing our PMI-Align method to SimAlign. They clearly show the advantages of using the PMI matrix over the similarity matrix. Both matrices are normalized with min-max normalization to be comparable.

C Number of Parameters and Runtimes

We use 3 pre-trained models in our experiments:

MBERT (Devlin et al., 2019), which is pre-trained with masked language modeling (MLM) and next sentence prediction on Wikipedia of 104 languages.

XLMR-base (Conneau et al., 2020), pre-trained with MLM on large-scale CommonCrawl data for 100 languages.

XLM-align (Chi et al., 2021), pre-trained with translation language modeling (TLM) and denoising word alignment (DWA) for 14 English-centric language pairs, along with MLM for 94 languages.

Our method has no parameters itself. However, considering the parameters of the used pre-trained LM, MBERT has about 170 million parameters, while XLMR-base and XLM-align both have about 270 million parameters.

Since our word aligner is simple and efficient, we did all our experiments on an Intel(R) Core(TM) i7-6700 CPU with 32GB memory and it just took about 0.1 seconds on average to align each parallel sentence in our whole dataset, while using XLMR-base model.

Table 2: Statistics and links for test datasets (Jalili Sabet et al., 2020)

Language pair	# of sentences	Link
En-Cs (Mareček, 2008)	2500	http://ufal.mff.cuni.cz/czech-english-manual-word-alignment
En-De	508	http://www-i6.informatik.rwth-aachen.de/goldAlignment
En-Fa (Tavakoli and Faili, 2014)	400	http://eceold.ut.ac.ir/en/node/940
En-Fr (Och and Ney, 2000)	447	http://web.eecs.umich.edu/~mihalcea/wpt
En-Hi	90	http://web.eecs.umich.edu/~mihalcea/wpt05
En-Ro (Mihalcea and Pedersen, 2003)	203	http://web.eecs.umich.edu/~mihalcea/wpt05

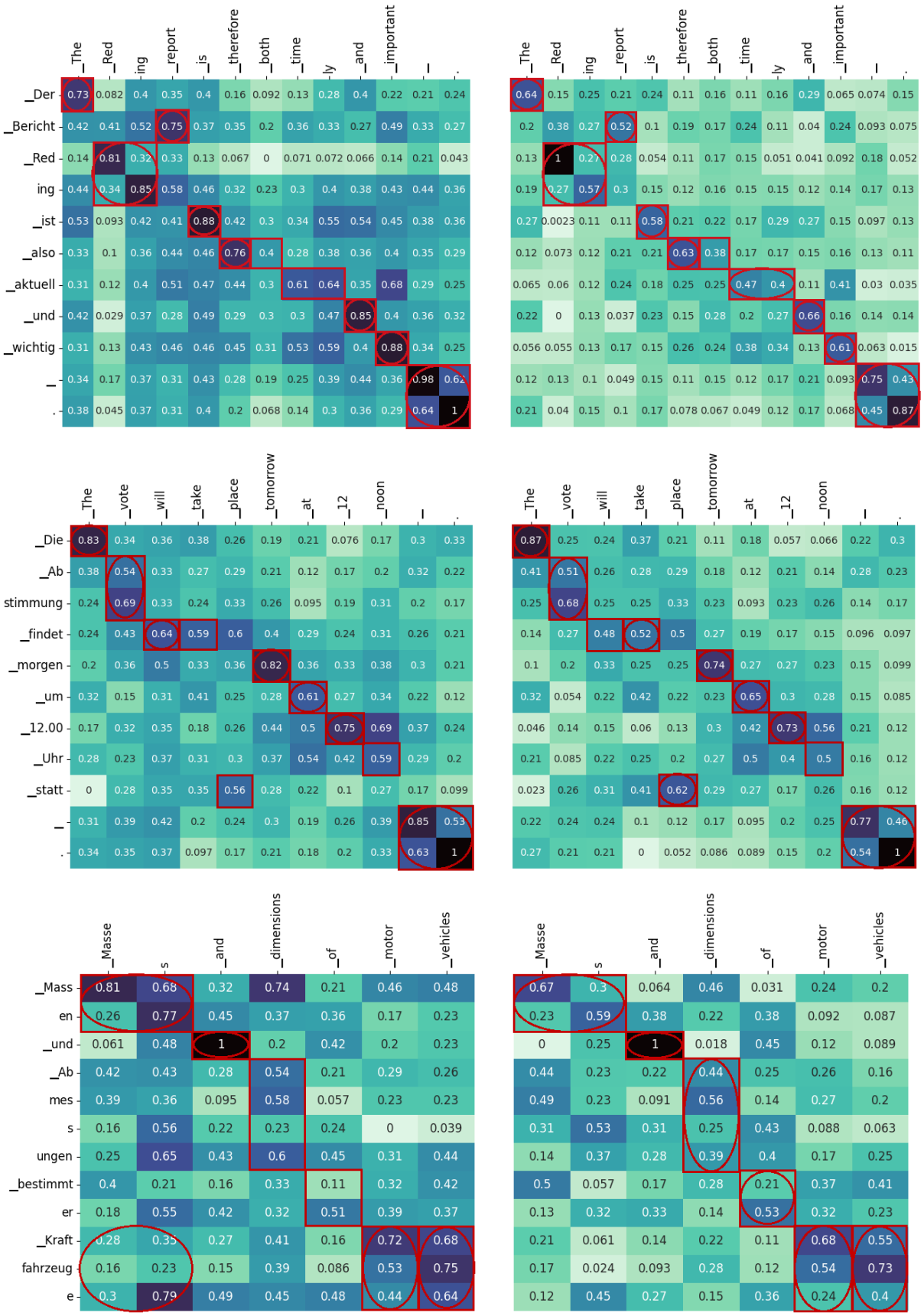


Figure 2: Similarity matrices (left) vs. PMI matrices (right) along with the word-level alignments extracted using SimAlign vs. PMI-Align for some parallel sentence pairs. Red boxes indicate the gold alignments, whereas red ovals show the aligners' outputs.

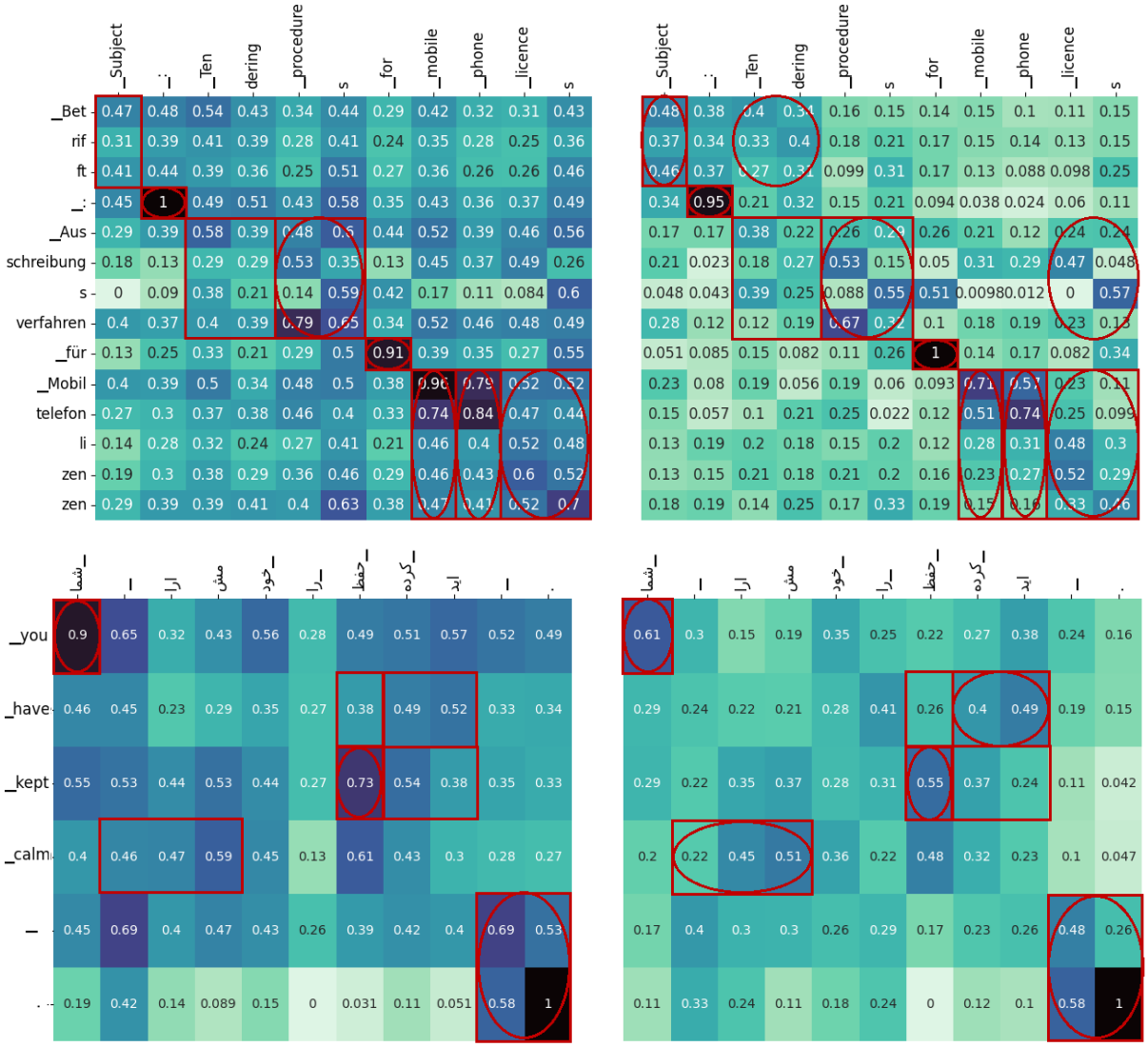


Figure 3: Additional examples.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section

☑ A2. Did you discuss any potential risks of your work?
Limitations section

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3, Appendix A

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data we used was a standard publicly available data used for the intended task in many prior works.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix A, C

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix A

C ☑ Did you run computational experiments?

Appendix C

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix C

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Not applicable. Our method doesn't have any hyperparameters

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Not applicable. Our results don't vary in different runs.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.