Embedding-Enhanced GIZA++: Improving Word Alignment Using Embeddings

Kelly Marchisio kmarc@jhu.edu
Department of Computer Science, Johns Hopkins University, Baltimore, MD, 21211, USA

Conghao Xiong[*] chxiong21@cse.cuhk.edu.hk
Department of Computer Science and Engineering, The Chinese University of Hong Kong,
New Territory, HKSAR

Philipp Koehn phi@jhu.edu
Department of Computer Science, Johns Hopkins University, Baltimore, MD, 21211, USA

Abstract

A popular natural language processing task decades ago, word alignment has been dominated until recently by GIZA++, a statistical method based on the 30-year-old IBM models. New methods that outperform GIZA++ primarily rely on large machine translation models, massively multilingual language models, or supervision from GIZA++ alignments itself. We introduce Embedding-Enhanced GIZA++, and outperform GIZA++ without any of the aforementioned factors. Taking advantage of monolingual embedding spaces of source and target language only, we exceed GIZA++'s performance in every tested scenario for three languages pairs. In the lowest-resource setting, we outperform GIZA++ by 8.5, 10.9, and 12 AER for Ro-En, De-En, and En-Fr, respectively. We release our code at `https://github.com/kellymarchisio/ee-giza`.

1 Introduction

Word alignment techniques were once ubiquitous in the machine translation (MT) literature, as they formed a critical part of statistical machine translation (SMT) systems. Since the advent of neural machine translation (NMT), word alignment is no longer a step in typical NMT training, but is still important for other tasks such as annotation transfer (e.g. Yarowsky and Ngai, 2001; Rasooli et al., 2018), as a post-processing step of MT to reinsert markup (e.g. Müller, 2017), and for some mapping-based unsupervised MT methods such as Artetxe et al. (2019).

GIZA++ (Och, 2003), a statistical alignment model, has been the most commonly used tool for word alignment quality for 20 years and is based the IBM translation models that are yet a decade older (Brown et al., 1993). Though a handful of neural systems have outperformed GIZA++, these rely on large MT models (e.g. Stengel-Eskin et al., 2019; Chen et al., 2020; Zenkel et al., 2020), massively multilingual language models (e.g. Garg et al., 2019b; Jalili Sabet et al., 2020; Dou and Neubig, 2021), supervision from human-annotated alignments (Nagata et al., 2020), or combinations of the above.

We introduce Embedding-Enhanced GIZA++ (EE-GIZA++), an improvement to GIZA++ without any of the aforementioned factors. EE-GIZA++ biases GIZA++ to align semantically similar words from a shared embedding space. We outperform GIZA++ in all tested settings

[*] Work completed at Johns Hopkins University.

on three language pairs. EE-GIZA++ is particularly strong in comparison with GIZA++ when parallel training data is scarce: using only $\sim$500 lines of bitext, it outperforms GIZA++ by 10.9 AER[1] and 12.0 AER for De-En and Fr-En, respectively.

2 Related Work

Fast-align is a statistical aligner similar to GIZA++. It is a reparameterization of IBM Model 2 (Dyer et al., 2013). eflomal is another highly-performant non-neural aligner (Östling and Tiedemann, 2016). We use GIZA++ as our base system because it commonly-used and trusted for generating high-quality alignments. Numerous improvements to GIZA++ have been proposed (e.g. Vaswani et al., 2012).

Recent work involves using neural translation models to guide or extract alignments, viewing attention as a proxy for alignment (e.g. Peter et al., 2017; Li et al., 2018; Garg et al., 2019b; Zenkel et al., 2019, 2020; Chen et al., 2020). Other aligners use massive multilingual language models with contextualized embeddings such as mBERT (Devlin et al., 2019). Like us, Jalili Sabet et al. (2020) experiment with mapped monolingual embedding spaces, but exceed the GIZA++ baseline only when using spaces such as mBERT and XLM-R (Conneau et al., 2020). Dou and Neubig (2021)'s approach is similar to the aforementioned authors, but they improve results by finetuning mBERT on auxiliary tasks. Nagata et al. (2020) use mBERT and require supervision with human-annotated alignments.

Pourdamghani et al. (2018) use word embedding similarity to augment parallel data seen by GIZA++, improving alignment and downstream low-resource MT. Jalili Sabet et al. (2016) also use nearest-neighbors in a word embedding space to alter IBM Model 1, but their performance does not match ours. Perhaps most similar to our work, Songyot and Chiang (2014) incorporate word similarity into GIZA++ using a feedforward neural network trained to model word similarity, with a hyperparameter to control the influence of the neural model.

3 Background

Let S be a source-language sentence of tokens $(s_1, s_2, ..., s_m)$ and T be a target-language sentence $(t_1, t_2, ..., t_l)$. Alignments are defined as $A \subseteq \{(s, t) \in S \times T\}$ where each s, t are meaningfully related—usually, translations of one another. Performance is typically measured with Alignment Error Rate (AER; Och and Ney, 2000a).

3.1 GIZA++

GIZA++ is a popular statistical alignment and MT toolkit (Och and Ney, 2000b, 2003) which implements IBM Models 1-5 (Brown et al., 1993) and the HMM Model (Vogel et al., 1996), trained using expectation-maximization (EM). The default training setup is to run five iterations each of IBM Model 1, HMM, Model 3, and Model 4. GIZA++ is highly effective at aligning frequent words in a corpus, but error-prone for infrequent words.

IBM Models The IBM models developed more than 30 years ago for MT are useful for alignment. IBM Model 1 relies on lexical translation probabilities $p(f|e)$ for source word e and target word f. Model 2 adds an alignment model $p(j \mid i, l, m)$, predicting source position j from target position i of sentences with lengths m and l, respectively. Model 3 adds a fertility model. Model 4 and the HMM Model replace the alignment with a relative reordering model. After training, the most likely alignment can be computed for a sentence pair.

[1] Alignment Error Rate (Och and Ney, 2000a).

3.2 Monolingual Embedding Space Mapping

Non-contextual vector representations of words ("word embeddings", "word vectors") are common in NLP (e.g. Mikolov et al., 2013; Bojanowski et al., 2017). Word vectors trained on monolingual data *embed* the word into an N-dimensional space where distance and angle have meaning. Mapping monolingual embedding spaces to a shared crosslingual space is common, particularly for bilingual lexicon induction and cross-lingual information retrieval.

Procrustes Problem Techniques that map monolingual embedding spaces to a crosslingual space often solve a variation of the generalized Procrustes problem (e.g., Artetxe et al., 2018b; Conneau et al., 2018; Patra et al., 2019; Ramírez et al., 2020). Given word embedding matrices $X, Y \in \mathbb{R}^{n\times d}$ where $x \in X$, $y \in Y$ are word vectors in source and target languages, one finds the map $W \in \mathbb{R}^{d\times d}$ that minimizes distances for each pair (x, y) known to be translations:

$$\arg\min_{W} \|XW - Y\|_F$$

When restricting W to be orthogonal ($WW^T = I$), Schönemann (1966) showed that the closed-form solution is $W = VU^T$, where $U\Sigma V$ is the singular value decomposition of Y^TX.

After mapping X and Y to a shared space with W, translations are extracted via nearest-neighbor search. A popular distance metric is cross-domain similarity local scaling (CSLS) to mitigate the "hubness problem" (Conneau et al., 2018).

4 Method

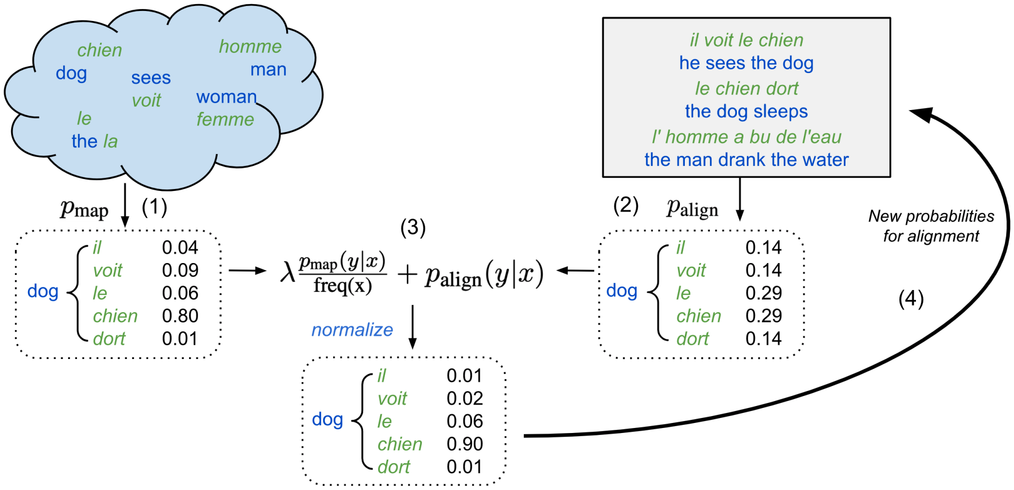

Figure 1: Proposed Method: Embedding-Enhanced GIZA++. 1) Map monolingual embeddings to crosslingual space. Calculate CSLS for cooccurring words and take softmax to calculate a probability distribution (*p_map*). 2) Use statistical aligner to calculate separate probability distribution over cooccuring words (*p_align*). 3) Interpolate distributions with weight proportional to source word's frequency. Normalize. 4) Replace the statistical model's translation probability table with updated probability distribution. 5) Repeat Steps 2-4 for each iteration of EM.

GIZA++ is highly effective at inducing the correct alignment for frequent words when parallel resources are abundant, but is error-prone for rare words. Because word embeddings can be trained on large amounts of monolingual data, rare words from a parallel corpus may be well-enough represented in a large monolingual corpus that reasonable word embeddings can be trained. Our key insight is that for infrequent words, finding a translation via nearest-neighbors in a shared embedding space may be more reliable than using a statistical aligner. We thus incorporate embedding space mapping into GIZA++ training, giving more or less influence to the statistical aligner depending on word frequency. Figure 1 shows the method.

1. Map embedding spaces. Word embedding spaces X and Y for source and target language, respectively, are mapped to a crosslingual space using VecMap[2] (Artetxe et al., 2018a).

2. Calculate translation probability distribution from mapped spaces. Let $\text{Co}_Y(x)$ be the words from the target language that cooccur with source word x in the corpus. For each x, we calculate a probability distribution over possible alignments from $\text{Co}_Y(x)$ with a softmax over the CSLS scores (We use $\tau = 0.1$.) We use the mapped embedding spaces for source and target languages to calculate CSLS.

$$p_{map}(y|x) = \frac{\exp\left(\text{CSLS}(x, y)/\tau\right)}{\sum\limits_{y' \in \text{Co}_Y(x)} \exp\left(\text{CSLS}(x, y')\right)/\tau)}$$

3. Integrate with GIZA++. Recall that IBM Models 1, 3, 4, and HMM maintain a lexical translation table of $p_{\text{align}}(y|x)$ for every cooccurring source-target word pair. During training of IBM Model 1 and the HMM, we interpolate the lexical translation table with embedding-based translation probabilities after each iteration of EM. For each cooccurring pair (x, y), calculate:

$$score(x, y) = \lambda \frac{p_{\text{map}}(y|x)}{\text{freq(x)}} + p_{\text{align}}(y|x)$$

where freq(x) is the raw frequency of x in the source-side of the corpus and λ is a hyperparameter. The effect is that p_{map} is given more weight for infrequent words, in accordance with our goal to trust the embedding space mapper for infrequent words and the statistical aligner for frequent words. Then normalize over cooocuring words:

$$p(y|x) = \frac{score(x, y)}{\sum\limits_{y_i \in \text{Co}_Y(x)} score(x, y_i)} \tag{1}$$

We update GIZA++'s lexical translation table with the new value from Equation 1 for all cooccurring pairs, then begin the next iteration of EM.[3] This process is repeated for all iterations of IBM Model 1 and HMM model training. IBM Model 3 and 4 are trained as usual. Integrating probabilites from p_{map} into IBM Models 3 and 4 is for future work.

Steps 1-3 are done in source→target and target→source directions. Alignments are symmetrized with grow-diag-final (Koehn et al., 2003).

5 Experimental Setup

We use the same training setup as previous work[4] (Garg et al., 2019b; Zenkel et al., 2019, 2020; Chen et al., 2020; Dou and Neubig, 2021). Training corpora for German-English (De-En), English-French (En-Fr), and Romanian-English (Ro-En) are 1.9M, 1.1M, and 448K lines, and test sets are 508, 447, and 248 lines, respectively. Validation sets do not exist, so we tune λ on

[2] github.com/artetxem/vecmap

[3] If a word from the bitext is not present in the word embedding space, its translation probability is not updated.

[4] https://github.com/lilt/alignment-scripts. Data: (Mihalcea and Pedersen, 2003; Koehn, 2005; Vilar et al., 2006)

1 million lines of De-En.[5] λ is set to 10,000. We use the VecMap implementation of CSLS and SciPy for some utility functions and softmax calculation (Virtanen et al., 2020; Harris et al., 2020). For pretrained word embedding spaces, we use the publicly-available Wikipedia word vectors trained using fastText from Bojanowski et al. (2017).[6] We limit vocabulary size to 200,000 and perform embedding mapping with VecMap in unsupervised mode.

	De-En		Ro-En		En-Fr	
Corpus Size	GIZA++	Ours	GIZA++	Ours	GIZA++	Ours
Test Set Only	44.2	**33.3** *(-10.9)*	42.8	**34.3** *(-8.5)*	26.9	**14.9** *(-12.0)*
1000	41.0	**31.1** *(-9.9)*	41.5	**33.6** *(-7.9)*	20.0	**11.4** *(-8.6)*
2000	37.7	**29.1** *(-8.6)*	39.6	**32.9** *(-6.7)*	17.2	**10.1** *(-7.1)*
5000	34.5	**26.9** *(-7.6)*	38.2	**32.0** *(-6.2)*	14.0	**8.5** *(-5.5)*
10,000	31.9	**25.5** *(-6.4)*	36.1	**30.4** *(-5.7)*	11.7	**7.5** *(-4.2)*
20,000	29.3	**24.2** *(-5.1)*	35.2	**30.3** *(-4.9)*	10.0	**7.1** *(-2.9)*
50,000	26.6	**22.6** *(-4.0)*	34.2	**29.7** *(-4.5)*	8.6	**6.3** *(-2.3)*
100,000	25.4	**21.9** *(-3.5)*	33.4	**29.3** *(-4.1)*	7.8	**6.1** *(-1.7)*
200,000	24.0	**21.2** *(-2.8)*	32.7	**29.4** *(-3.3)*	7.0	**5.8** *(-1.2)*
500,000	21.6	**20.3** *(-1.3)*	26.5	**25.5** *(-1.0)*	6.1	**5.7** *(-0.4)*
1,000,000	20.7	**20.1** *(-0.6)*	*n/a*	*n/a*	6.1	**5.5** *(-0.6)*
1,900,000	20.6	**19.9** *(-0.7)*	*n/a*	*n/a*	*n/a*	*n/a*

Table 1: Main Results. Alignment Error Rate (AER) of EE-GIZA++ vs. GIZA++ baseline (lower is better). Test set is included in corpus size. Ro-En 500K is the full 448K training set. Bidirectional, symmetrized (grow-diag-final).

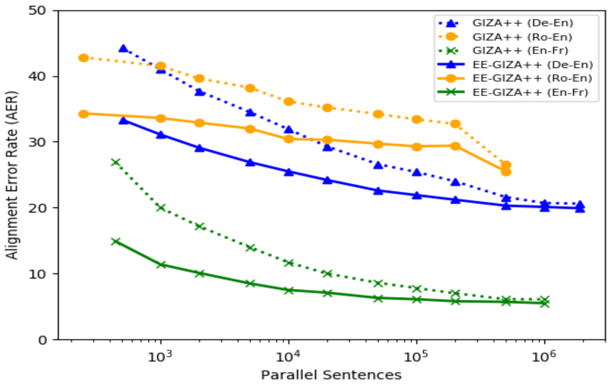

Figure 2: Visualization of Main Results. Alignment Error Rate (AER) of EE-GIZA++ vs. GIZA++ baseline for increasing amounts of training data. Lower is better.

[5] This was the approximate average size of training data for all languages.

[6] `https://fasttext.cc/docs/en/pretrained-vectors.html`

6 Results

The main results are presented in Table 1 and visualized in Figure 2. We observe that EE-GIZA++ consistently outperforms GIZA++ by a large margin in every tested scenario. When aligning the test set alone with no additional bitext, EE-GIZA++ dramatically outperforms GIZA++: by 8.5 AER for Ro-En, 10.9 AER for De-En, and 12 AER for En-Fr. This represents improvements of approximately 20%, 25%, and 45% for Ro-En, De-En, and En-Fr, respectively. The error-rate improvement is especially notable when we consider that each test set has only approximately 250-500 lines. When expanding the training set to include a total of 10,000 lines, we continue to observe strong gains with our method: with absolute improvements of 5.7, 6.4, and 4.2 AER for Ro-En, De-En, and En-Fr. These represent improvements of approximately 15.8%, 20.1%, and 35.9%, respectively.

Statistical Baselines	**De-En**	**Ro-En**	**En-Fr**
GIZA++	20.6	26.5	6.2
eflomal*	22.6	25.1	8.2
fast-align*	27.0	32.1	10.5
Massively-Multilingual			
Jalili Sabet et al. (2020)	19.†	27.2*[7]	6.†
Dou and Neubig (2021)	15.6	23.0	4.4
no fine-tuning	17.4	27.9	5.6
Bilingual NMT-Based			
Zenkel et al. (2019)	21.2	27.6	10.0
Garg et al. (2019b)	20.2	26.0	7.7
using GIZA++ output	16.0	23.1	4.6
Zenkel et al. (2020)	16.3	23.4	5.0
Chen et al. (2020)	15.4	21.2	4.7
Ours	19.9	25.5	5.3

Table 2: Supplemental results in high-resource settings compared to models that use additional resources. “Massively multilingual” models use mBERT. NMT models likely fail in low-bitext scenarios (our focus). Bidirectional. *reported in Dou and Neubig (2021). †Jalili Sabet et al. (2020) report one less significant digit.

Supplemental Results: High-Resource We use the full data sets for De-En, Ro-En, and En-Fr and compare to existing work in Table 2.[8] We outperform the three statistical baselines, except eflomal on Ro-En. EE-GIZA++ outperforms Jalili Sabet et al. (2020) on Ro-En and En-Fr, which utilizes a massively-multilingual language model. Dou and Neubig (2021) with fine-tuning outperforms our model, though they use mBERT which is trained on 104 languages. Notably, Garg et al. (2019a) use GIZA++ output as supervision. EE-GIZA++ performs better than GIZA++, so AER might improve if supervised with our alignments.

[7] As Jalili Sabet et al. (2020) use the 2005 Ro-En test set from `https://web.eecs.umich.edu/~mihalcea/wpt05`, we report Dou and Neubig (2021)’s Ro-En results here for consistency with the others, which use the 2003 test set (`https://web.eecs.umich.edu/~mihalcea/wpt`.

[8] Many of these use the grow-diag symmetrization heuristic, but we use grow-diag-final.

7 Conclusion and Future Work

We introduce EE-GIZA++, an unsupervised enhancement to GIZA++ that uses word embeddings for improved word alignment in low-bitext settings, without the use of NMT or massively-multilingual language models that to-date have been the strongest competitors to GIZA++. EE-GIZA++ outperforms GIZA++ by 8.5, 10.9, and 12 AER in lowest-bitext scenarios for Ro-En, De-En, and En-Fr, respectively. Future work should examine performance of EE-GIZA++ on a diverse set of languages with varying scripts and amounts of data available.

Acknowledgements

The authors would like to thank Elias Stengel-Eskin, Mahsa Yarmohammadi, and Marc Marone for fruitful discussion on alignment. The first author would like to thank Martin St. Denis for his ideas about the graphic used in this work.

References

Artetxe, M., Labaka, G., and Agirre, E. (2018a). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Artetxe, M., Labaka, G., and Agirre, E. (2018b). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (2019). An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy. Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chen, Y., Liu, Y., Chen, G., Jiang, X., and Liu, Q. (2020). Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2018). Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dou, Z.-Y. and Neubig, G. (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, Online. Association for Computational Linguistics.

Dyer, C., Chahuneau, V., and Smith, N. A. (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, Atlanta, Georgia. Association for Computational Linguistics.

Garg, S., Moniz, J. R. A., Aviral, A., and Bollimpalli, P. (2019a). Learning to relate from captions and bounding boxes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6597–6603, Florence, Italy. Association for Computational Linguistics.

Garg, S., Peitz, S., Nallasamy, U., and Paulik, M. (2019b). 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, Hong Kong, China. Association for Computational Linguistics.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585:357–362.

Jalili Sabet, M., Dufter, P., Yvon, F., and Schütze, H. (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, Online. Association for Computational Linguistics.

Jalili Sabet, M., Faili, H., and Haffari, G. (2016). Improving word alignment of rare words with word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3209–3215, Osaka, Japan. The COLING 2016 Organizing Committee.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Li, X., Liu, L., Tu, Z., Shi, S., and Meng, M. (2018). Target foresight based attention for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1380–1390.

Mihalcea, R. and Pedersen, T. (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.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Müller, M. (2017). Treatment of markup in statistical machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46, Copenhagen, Denmark. Association for Computational Linguistics.

Nagata, M., Chousa, K., and Nishino, M. (2020). A supervised word alignment method based on cross-language span prediction using multilingual BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 555–565, Online. Association for Computational Linguistics.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2000a). Improved statistical alignment models. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 440–447.

Och, F. J. and Ney, H. (2000b). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Östling, R. and Tiedemann, J. (2016). Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics*, pages 125–146.

Patra, B., Moniz, J. R. A., Garg, S., Gormley, M. R., and Neubig, G. (2019). Bilingual lexicon induction with semi-supervision in non-isometric embedding spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 184–193, Florence, Italy. Association for Computational Linguistics.

Peter, J.-T., Nix, A., and Ney, H. (2017). Generating alignments using target foresight in attention-based neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1):27–36.

Pourdamghani, N., Ghazvininejad, M., and Knight, K. (2018). Using word vectors to improve word alignments for low resource machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 524–528, New Orleans, Louisiana. Association for Computational Linguistics.

Ramírez, G., Dangovski, R., Nakov, P., and Soljačić, M. (2020). On a novel application of wasserstein-procrustes for unsupervised cross-lingual learning. *arXiv preprint arXiv:2007.09456*.

Rasooli, M. S., Farra, N., Radeva, A., Yu, T., and McKeown, K. (2018). Cross-lingual sentiment transfer with limited resources. *Machine Translation*, 32(1):143–165.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Songyot, T. and Chiang, D. (2014). Improving word alignment using word similarity. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1840–1845, Doha, Qatar. Association for Computational Linguistics.

Stengel-Eskin, E., Su, T.-r., Post, M., and Van Durme, B. (2019). A discriminative neural model for cross-lingual word alignment. 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 910–920, Hong Kong, China. Association for Computational Linguistics.

Vaswani, A., Huang, L., and Chiang, D. (2012). Smaller alignment models for better translations: Unsupervised word alignment with the l0-norm. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 311–319, Jeju Island, Korea. Association for Computational Linguistics.

Vilar, D., Popović, M., and Ney, H. (2006). Aer: Do we need to "improve" our alignments? In *International Workshop on Spoken Language Translation (IWSLT) 2006*.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Vogel, S., Ney, H., and Tillmann, C. (1996). Hmm-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Yarowsky, D. and Ngai, G. (2001). Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Zenkel, T., Wuebker, J., and DeNero, J. (2019). Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*.

Zenkel, T., Wuebker, J., and DeNero, J. (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, Online. Association for Computational Linguistics.