Word Alignment by Fine-tuning Embeddings on Parallel Corpora

Zi-Yi Dou, Graham Neubig

EACL 2021

Overview

- Most of previous work (e.g., GIZA++ (2003), fastalign (2013)) on word alignment has worked by performing unsupervised learning on parallel text.
- Recent work (SimAlign – 2020) has shown competitive word alignment performance with statistical models by using pretrained multilingual language models, even without finetuning.
- This work (AwesomeAlign) proposes objectives to combine the strength of (i) finetuning on parallel text and (i) pretrained multilingual language models.

Word Alignment

- $\mathbf{x} = \langle x_1, \cdots, x_n \rangle$ is a sentence in the source language.
- $\mathbf{y} = \langle y_1, \cdots, y_m \rangle$ is the translated sentence for $\mathbf{x}$ in the target language.
- Find a set: $A = \{\langle x_i, y_j \rangle : x_i \in \mathbf{x}, y_j \in \mathbf{y}\}$

Such that for each pair $\langle x_i, y_j \rangle$, x_i and y_j are semantically similar <u>within</u> the context of the sentence.

Proposed Method: Similarity Matrix Computation

- Extracting word embeddings: via a multilingual language model (e.g., mBERT, XLM-Roberta):

$$h_{\mathbf{x}} = \langle h_{x_1}, \cdots, h_{x_n} \rangle$$

source sentence

$$h_{\mathbf{y}} = \langle h_{y_1}, \cdots, h_{y_m} \rangle$$

target sentence

- 2 ways to compute similarity matrix:

> Using cosine similarity: $S = h_{\mathbf{x}} h_{\mathbf{y}}^T$; $S_{\mathbf{xy}} = \mathcal{N}(S)$
where $\mathcal{N}(\ )$ is a normalization function such as: *softmax, sparsemax*.

> Using optimal transport:
+ Each sentence is a set of points (i.e., words).
+ Assumption: each word/point is uniformly distributed.
+ Distance function $C(x_i, y_j)$: cosine, Euclidean, dot product.
+ Using Sinkhorn-Knopp to compute the transition matrix $S_{\mathbf{xy}}$ via minimizing:

$$\sum_{i,j} C(x_i, y_j) S_{\mathbf{xy}_{ij}}$$

Proposed Method: Word Alignment Deduction

- Once we computed the transition matrix $S_{\mathbf{xy}}$, the final alignment is obtained by:

$$A = (S_{\mathbf{xy}} > c) * (S_{\mathbf{yx}}^T > c)$$

Where c is a probability threshold (set to 0.001 in this work); $S_{\mathbf{xy}}$ represents for alignment probabilities from source to target sentence while $S_{\mathbf{yx}}^T$ represents for alignment probabilities from target to source sentence.

Proposed Method: Further Improvement with Finetuning Objectives

- **Masked Language Modeling**: requires monolingual corpora for two languages.

$$L_{MLM} = \log p(\mathbf{x}|\mathbf{x}^{mask}) + \log p(\mathbf{y}|\mathbf{y}^{mask})$$

- **Translation Language Modeling**: requires parallel sentences for two languages.

$$L_{TLM} = \log p([\mathbf{x};\mathbf{y}]|[\mathbf{x}^{mask};\mathbf{y}^{mask}]) + \log p([\mathbf{y};\mathbf{x}]|[\mathbf{y}^{mask};\mathbf{x}^{mask}]).$$

- **Self-training Objective**: requires parallel sentences for two languages.

$$L_{SO} = \sum_{i,j} A_{ij} \frac{1}{2}\left(\frac{S_{\mathbf{xy}_{ij}}}{n} + \frac{S_{\mathbf{yx}_{ij}}}{m}\right)$$ where A is obtained by the non-finetuning method.

- **Parallel Sentence Identification**: requires parallel sentences for two languages. Feeding [CLS] vector to a feedforward net to product $s(\mathbf{x}', \mathbf{y}')$ for predicting whether two sentences are parallel.

$$L_{PSI} = l \log s(\mathbf{x}', \mathbf{y}') + (1-l)\log(1 - s(\mathbf{x}', \mathbf{y}'))$$

- **Consistency Optimization**: requires parallel sentences for two languages. Encouraging the agreement between the two alignment directions (i.e., source-to-target and target-to-source):

$$L_{CO} = -\frac{\mathrm{trace}(S_{\mathbf{xy}}^{\mathrm{T}} S_{\mathbf{yx}})}{\min(m,n)}$$

- **Minimizing the overall loss function**: $L = L_{MLM} + L_{TLM} + L_{SO} + L_{PSI} + \beta L_{CO}$

Results:

Model		Setting	De-En	Fr-En	Ro-En	Ja-En	Zh-En
Baseline							
SimAlign	(2020)	*w/o fine-tuning*	18.8	7.6	27.2	46.6	21.6
fast_align	(2013)	*bilingual*	27.0	10.5	32.1	51.1	38.1
eflomal	(2016)	*bilingual*	22.6	8.2	25.1	47.5	28.7
GIZA++	(2003)	*bilingual*	20.6	5.9	26.4	48.0	35.1
Zenkel et al. (2020)		*bilingual*	16.0	5.0	23.4	-	-
Chen et al. (2020)		*bilingual*	15.4	4.7	21.2	-	-
Ours							
α-entmax		*w/o fine-tuning*	18.1	5.6	29.0	46.3	18.4
		bilingual	16.1	***4.1***	23.4	38.6	15.4
		multilingual (β = 0)	15.4	***4.1***	22.9	***37.4***	**13.9**
		multilingual (β = 1)	***15.0***	4.5	**20.8**	38.7	14.5
		zero-shot	16.0	4.3	28.4	44.0	**13.9**
softmax		*w/o fine-tuning*	17.4	5.6	27.9	45.6	18.1
		bilingual	15.6	**4.4**	23.0	38.4	15.3
		multilingual (β = 0)	15.3	**4.4**	22.6	**37.9**	***13.6***
		multilingual (β = 1)	**15.1**	4.5	***20.7***	38.4	14.5
		zero-shot	15.7	4.6	27.2	43.7	14.0

Results:

	Component	De-En	Fr-En	Ro-En	Ja-En	Zh-En	Speed
Prob.	*softmax*	**17.4**	**5.6**	**27.9**	**45.6**	**18.1**	**33.22**
	α-entmax	18.1	**5.6**	29.0	46.3	18.4	32.36
OT	Cosine	24.4	15.7	33.7	54.0	31.1	3.36
	Dot Product	25.4	17.1	34.1	54.2	30.9	3.82
	Euclidean	20.7	15.1	33.3	53.2	29.8	3.05

Model	Layer	De-En	Fr-En	Zh-En
mBERT	7	18.7	6.1	19.1
	8	**17.4**	**5.6**	***18.1***
	9	18.8	6.1	20.1
XLM-15 (MLM)	4	21.1	6.8	**25.3**
	5	**20.4**	**6.1**	26.1
	6	23.2	7.7	33.3
XLM-15 (MLM+TLM)	4	16.4	4.9	**18.6**
	5	***16.2***	***4.7***	23.7
	6	18.8	5.7	26.2
XLM-100 (MLM)	7	20.5	8.5	30.8
	8	**19.8**	**8.2**	**28.6**
	9	19.9	8.8	29.3
XLM-R	5	24.4	10.3	33.2
	6	**23.1**	**9.2**	30.7
	7	24.7	11.5	**28.1**