Fixing Translation Divergences in Parallel Corpora for Neural MT

Anonymous EMNLP submission

Abstract

Corpus-based approaches to machine translation rely on the availability of parallel corpora. According to the process followed to compile a parallel corpus, it may contain multiple parallel sentences that are often not as parallel as one might assume. This paper describes an unsupervised method for detecting translation divergences in parallel sentences. We present a neural network to predict sentence similarity that minimises a loss function based on word alignments. We show that accurate predictions are obtained allowing divergent sentences to be filtered out. Furthermore, word similarity scores predicted by the network are used to identify and fix some divergences guiding to more parallel segments. We evaluate the presented method on an English-French and an English-German machine translation tasks. Results show that neural MT systems trained on the filtered/corrected corpus outperform the MT systems trained on the original data.

1 Introduction

Parallel sentence pairs are the only necessary resource to build Machine Translation (MT) systems. In the case of Neural MT, a large neural network is trained through maximising translation performance on a given parallel corpus. Therefore, the quality of an MT engine is heavily dependent upon the amount and quality of parallel sentences. Unfortunately, parallel texts are scarce resources. There are relatively few language pairs for which parallel corpora of reasonable sizes are available, and even for those pairs, the corpora come mostly from few domains. To alleviate the lack of parallel data several approaches have been developed in the last years. They range from methods using non-parallel, or comparable data (Zhao and Vogel, 2002; Fung and Cheung, 2004; Munteanu and Marcu, 2005; Grégoire and Langlais, 2017; Grover and Mitra, 2017) to techniques that produce synthetic parallel data from monolingual corpora (Sennrich et al., 2016a; Chinea-Rios et al., 2017) using in all cases automated alignment/translation engines that are prone to the introduction of noise in the resulting parallel sentences. Mismatches on parallel sentences extracted from translated texts are also reported (Tiedemann, 2011; Xu and Yvon, 2016). This problem is mostly ignored in machine translation, where parallel sentences are considered to convey the exact same meaning, and is particularly important for neural MT engines as suggested by (Chen et al., 2016).

Table 1 illustrates some examples of English-French parallel sentences which are not completely semantically equivalent. Examples are extracted from the OpenSubtitles corpus (Lison and Tiedemann, 2016). Divergences are outlined using bold letters. Additional segments are included on either side of the parallel sentences (first and second rows). Some translations may be completely uncorrelated (third row) or inaccurate (fourth row). Note that divergent translations are due to many different reasons (Li et al., 2014), the study of which is beyond the scope of this paper.

en	*What do you feel*, **Spock***?*
fr	*Que ressentez-vous?*
gl	*What do you feel?*
en	*How much do you get paid?*
fr	*T'es payé combien* **de l'heure***?*
gl	*How much do you get paid per hour?*
en	**That seems a lot.**
fr	**40 livres?**
gl	*40 pounds?*
en	*I brought you* **french fries***!*
fr	*Je t'ai rapporté des* **saucisses***!*
gl	*I brought you sausage!*

Table 1: Examples of semantically divergent parallel sentences. English (en), French (fr) and gloss of French (gl).

In this work, we present a neural divergence classifier aimed at detecting words on either side of parallel sentences for which the corresponding meaning is not present on the translated counterpart. We evaluate the classifier on an English-French and an English-German translation tasks showing that translation accuracy can be improved by filtering out divergent sentence pairs. In addition, we show that some divergent sentences can be fixed by removing divergent words, further boosting translation accuracy.

The remainder of this paper is structured as follows. Section 2 overviews related work. We describe in detail the core of the neural divergence classifier and the correction algorithm in Section 3. Section 4 details the experiments conducted to measure the ability of the presented classifier to identify and fix divergent sentences. Section 5 evaluates results. Finally, conclusions are drawn in Section 6 while further work is outlined in Section 7. All the code used in this paper as well as a human annotated test set are freely available[1].

2 Related Work

Attempts to measure the impact of translation divergences in MT systems have focused on the introduction of noise in sentence alignments (Goutte et al., 2012), showing that statistical MT systems are highly tolerant to noise, and that performance only degrades seriously at very high noise levels. In constrast, neural MT engines seem to be more sensitive (Chen et al., 2016), as they tend to assign high probabilities to rare events (Hassan et al., 2018).

Efforts have been devoted to characterise the degree of semantic equivalence between two snippets of text in the same or different languages (Agirre et al., 2016), a workshop devoted to an objective similar to our work. In (Mueller and Thyagarajan, 2016), a monolingual sentence similarity network is proposed, making use of a simple LSTM layer to compute sentence representations. The authors show that a simple SVM classifier can be built on top of the sentence representations to achieve state-of-the-art results in a semantic entailment classification task. With the same objective, the system presented in (He and Lin, 2016) uses multiple convolutional layers and models pairwise word interactions.

Our work is inspired by (Carpuat et al., 2017) where the authors train a cross-lingual divergence detector using word alignments and sentence length features to train a linear SVM classifier. Their work shows that an NMT system trained only on non-divergent sentences yields slightly higher translation quality scores and requires clearly less training time. The same authors have recently updated their work in (Vyas et al., 2018). The objective is the same as the neural network presented in (He and Lin, 2016) and their network further outperforms their previous work. Our work differs from the previous as we make use of a network with different topology. We model sentence similarity by means of optimising a loss function based on word alignments. Furthermore, the network predicts word similarity scores that we further use to correct divergent sentences.

3 Neural Divergence Classifier

The architecture of our model is inspired by the work on Word Alignment in (Legrand et al., 2016). Figure 1 illustrates the model.

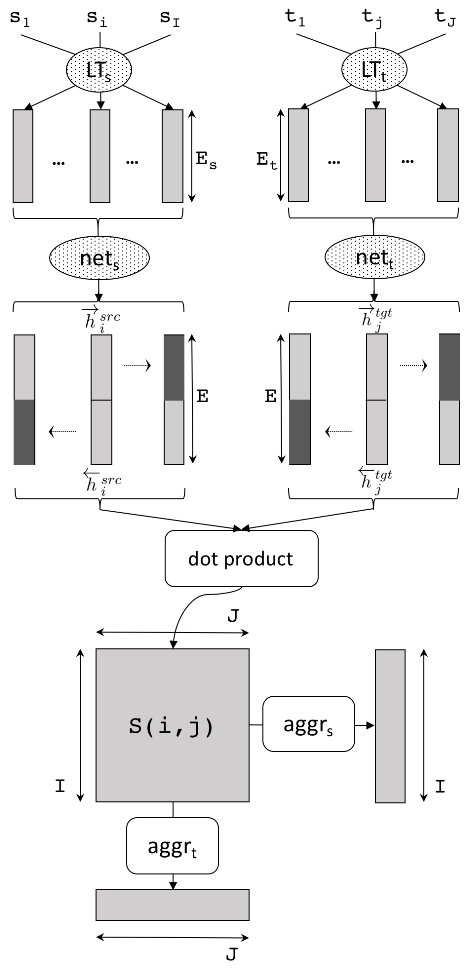

s_1
s_i
s_I
t_1
t_j
t_J
LT_s
LT_t
...
E_s
E_t
net_s
net_t
$\overrightarrow{h}_i^{src}$
$\overrightarrow{h}_j^{tgt}$
E
E
$\overleftarrow{h}_i^{src}$
$\overleftarrow{h}_j^{tgt}$
dot product
J
S(i,j)
$aggr_s$
I
I
$aggr_t$
J

Figure 1: Illustration of the model.

In the following, we consider a source-target sentence pair (s, t) with $s = (s_1, ..., s_I)$ and $t = (t_1, ..., t_J)$. The model is composed of 2

[1] https://github.com/anonymised

Bi-directional LSTM subnetworks, net_s and net_t, which respectively encode source and target sentences. Since both net_s and net_t take the same form we describe only the source architecture.

The source-sentence Bi-LSTM network outputs forward and backward hidden states, $\overrightarrow{h}_i^{src}$ and $\overleftarrow{h}_i^{src}$, which are then concatenated into a single vector encoding the i^{th} word of the source sentence, $h_i^{src} = [\overrightarrow{h}_i^{src}; \overleftarrow{h}_i^{src}]$. In addition, the last forward/backward hidden states (outlined using dark grey in Figure 1) are also concatenated into a single vector to represent whole sentences $h_{src} = [\overrightarrow{h}_I^{src}; \overleftarrow{h}_1^{src}]$. At this point, a measure of similarity between sentence pairs can be obtained by cosine similarity:

$$sim(h_{src}, h_{tgt}) = \frac{h_{src} \cdot h_{tgt}}{||h_{src}|| * ||h_{tgt}||} \qquad (1)$$

Our model can be optimised following two different objective functions:

wemb Maximise the word alignment scores between words of both sentences using aggregation functions that summarise the alignment scores for each source/target word. Similar to (Legrand et al., 2016) alignment scores $s(i, j)$ are given by the dot-product $S(i, j) = h_i^{src} \cdot h_j^{tgt}$, while aggregation functions are defined as:

$$\begin{aligned} aggr_s(i, S) &= \frac{1}{r}\, log \left(\sum_{j=1}^{J} e^{r*S(i,j)} \right) \\ aggr_t(j, S) &= \frac{1}{r}\, log \left(\sum_{i=1}^{I} e^{r*S(i,j)} \right) \end{aligned} \qquad (2)$$

The loss function is defined as:

$$\begin{aligned} &\mathcal{L}(src, tgt) = \\ &\quad \sum_{i=1}^{I} log(1 + e^{s_{aggr}(i,tgt)*sign_i}) + \\ &\quad + \sum_{j=1}^{J} log(1 + e^{s_{aggr}(src,j)*sign_j}) \end{aligned} \qquad (3)$$

semb Maximise the similarity score of both sentence embeddings as computed by Equation 1. With loss function defined as:

$$\mathcal{L}(src, tgt) = log(1+e^{sim(src,tgt)*sign}) \qquad (4)$$

3.1 Training with Negative Examples

Training is performed by minimising Equation 3, for which examples with annotations for source $sign_i$ and target $sign_j$ words are needed. As positive examples we use **(P)**aired sentences of a parallel corpus. All words in paired sentences are labelled as parallel (-1). As negative examples we use random **(U)**npaired sentences. In this case, all words are labelled as divergent $(+1)$. Since negative pairs may be very easy to classify and we want our network to detect less obvious divergences, we further create negative examples following:

We **(R)**eplace random sequences of words by a sequence of words with same parts-of-speech. The rationale behind this method is to keep the new sentences as grammatical as possible. Words that are not replaced are considered parallel (-1) while those replaced are assigned the divergent label $(+1)$. Words aligned to any replaced word are also assigned the divergent label $(+1)$. For instance, given the original sentence pair:

src:	What	do	you	feel	?
tgt:	Que	ressentez-vous	?		

We may replace ’you feel’, with tags ’PRP VB’, by another with same tags (i.e. ’we want’):

src:	What	do	**we**	**want**	?
$sign_i$:	-1	-1	**+1**	**+1**	-1
tgt:	Que	**ressentez-vous**	?		
$sign_j$:	-1	**+1**	-1		

Word alignments are used to identify ’ressentez-vous’ as divergent, since it is aligned to ’you feel’.

Motivated by sentence segmentation errors observed in many corpora, we also build negative examples by **(I)**nserting a second sentence following/preceding an original sentence. Only inserted words are considered divergent $(+1)$. Following with our example, we may add the sentence ’Not .’ at the end of the original source sentence:

src:	What	do	you	want	?	**Not**	**.**
$sign_i$:	-1	-1	-1	-1	-1	**+1**	**+1**
tgt:	Que	ressentez-vous	?				
$sign_j$:	-1	-1	-1				

Replace and insert methods are applied on either side of the sentence pairs. To avoid that negative examples are easily predicted by using the difference in length of sentences, we restrict negative examples to have a difference in length not exceeding a ratio of 2.0 (3.0 for shortest sentences).

The loss function of Equation 4 uses a single label for each sentence pair. Positive examples are labelled $sign = -1$ while any of the previous negative examples are labelled $sign = +1$.

3.2 Divergence Correction

We observed in our training corpora that many divergent sentences follow a common pattern, consisting of adding some extra leading/trailing words. Accordingly, we implemented a simple algorithm that discards sequences of leading/trailing words on both sides. Hence considering as parallel s_u^v and t_x^y. To find the optimal source (u, v) and target (x, y) indexes that enclose the parallel segments within the original sentence, we implement:

$$\arg\max_{u,v,x,y} \Big\{ \sum_{u \leq I \leq v} \max_{x \leq j \leq y} \{S(i,j)\} \Big\}$$

The $\mathcal{N}$-best sequences following the previous function (s_u^v, t_x^y) are considered as valid corrections, but only the highest ranked according to their similarity score is used as replacement for the original (s_1^I, t_1^J). Short sentences are not considered. This is, $v - u > \tau$ and $y - x > \tau$. Figure 2 (left) shows an example of an alignment matrix $S(i,j)$ for a given sentence pair. An acceptable correction is: *Que ressentez-vous ?* ⇔ *What do you feel ?*. Hence, with indexes $u = 1$, $v = 5$, $x = 1$ and $y = 3$.

4 Experiments

4.1 Corpora

We filter out divergences found in the English-French OpenSubtitles corpus (Lison and Tiedemann, 2016), which consists of a collection of movie and TV subtitles. We also use the English-German Paracrawl[2] corpus. Both corpora present many potential divergences. To evaluate English-French performance we use the En-Fr Microsoft Spoken Language Translation corpus, created from actual conversations over Skype (Federmann and Lewis, 2016). English-German performance is evaluated on the publicly available newstest-2017 En-De test set, corresponding to news stories selected from online sources (Bojar et al., 2017). In order to better assess the quality of our classifier when facing different types of word divergences we collected and annotated at the word level 500 sentences from the original OpenSubtitles corpus, containing: 200 paired sentences; 100 unpaired sentences; 100 sentences with replace examples; and 100 sentences with insert examples as detailed in Section 3.1.

4.2 Neural Divergence

All data is preprocessed with an in-house toolkit that performs minimal tokenisation. After tokenisation, each out-of-vocabulary word is mapped to a special UNK token. Vocabularies consist of the $50,000$ more frequent words. Word embeddings are initialised using `fastText`[3]. Size of embeddings is $E_s = E_t = 256$ cells. Both Bi-LSTM use 256-dimensional hidden representations ($E = 512$). We use $r = 1.0$. Network optimization is done using the SGD method along with gradient clipping (Pascanu et al., 2013). For each epoch we randomly select 1 million sentence pairs that we place in batches of 32 examples. We run 10 epochs and start decaying at each epoch by 0.8 when the loss on validation set increases. Two versions of the network are evaluated. The first optimises the loss shown by equation 3, the second optimises the loss shown by equation 4. Divergence is always computed following equation 1. For divergence correction, we used $\mathcal{N} = 20$ and $\tau = 3$.

4.3 Neural Translation

In addition to the basic tokenisation detailed in Section 4.2 we perform Byte-Pair Encoding (Sennrich et al., 2016b) with $30,000$ merge operations learned from both English and French data. Neural MT systems are based on the open-source project `OpenNMT`[4]. We use a Transformer model with same configuration in paper (Vaswani et al., 2017), i.e, both encoder and decoder have 6 layers; Multi-head attention is performed over 8 heads. Hidden layer's size is 512. The inner layer of feed-forward network is of size 2048. Word embeddings have a size of 512 cells. We set the dropout probability to 0.1. Batch size is set to 3072. The maximum length of both source and target sentences is set to 80 and we limit the vocabulary size to $50K$ words for both source and target languages.

The optimiser is Adam avec learning rate that was introduced in paper (Vaswani et al., 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\eta = 10^{-9}$, $warmup\_steps = 4000$. We stop training after 30 epochs.

[2] http://paracrawl.eu/

[3] https://github.com/facebookresearch/fastText

[4] http://opennmt.net

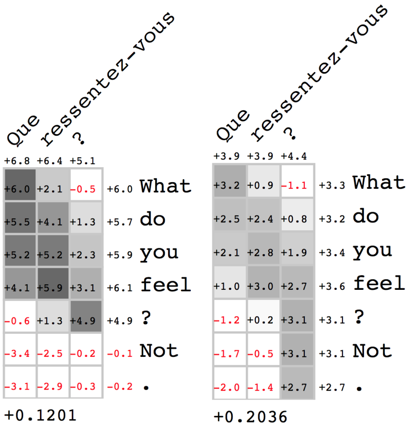
5 Results

Table 2 shows accuracies obtained by our model when trained over different combinations of negative examples. We use the test set with manual annotations of word divergences. In training, the same number of examples are always generated for each type of example (P, U, R and I). A word is considered divergent if it has a negative aggregation score, Equation 2.

Accuracy		Test examples				
		P	U	R	I	PURI
Train examples	PU	**0.996**	**0.994**	0.671	0.673	0.874
	PR	**0.995**	0.033	**0.951**	0.689	0.746
	PI	**0.998**	0.071	0.697	**0.725**	0.705
	PUR	**0.994**	**0.989**	**0.919**	0.710	0.932
	PUI	**0.995**	**0.996**	0.662	**0.769**	0.887
	PRI	**0.991**	0.161	**0.924**	**0.719**	0.768
	PURI	**0.995**	**0.980**	**0.916**	**0.788**	**0.942**

Table 2: Word divergence accuracies according to different type of examples used in train/test.

As it can be seen, non-divergent words in parallel and unpaired sentences (columns P and U) are easy to identify, as far as the model has seen these types of examples in training. Regarding columns R and I, accuracies are lower since these sentences contain a mix of divergent and non-divergent words. Again, models that were trained with the corresponding examples obtain the highest accuracies (outlined in bold letters). Column PURI shows results over the entire test set, mixing all type of examples. The best accuracy is obtained by the system trained on all type of examples.



	Que	ressentez-vous	?		
	+6.8	+6.4	+5.1		
	+6.0	+2.1	-0.5	+6.0	What
	+5.5	+4.1	+1.3	+5.7	do
	+5.2	+5.2	+2.3	+5.9	you
	+4.1	+5.9	+3.1	+6.1	feel
	-0.6	+1.3	+4.9	+4.9	?
	-3.4	-2.5	-0.2	-0.1	Not
	-3.1	-2.9	-0.3	-0.2	.
+0.1201					

	Que	ressentez-vous	?		
	+3.9	+3.9	+4.4		
	+3.2	+0.9	-1.1	+3.3	What
	+2.5	+2.4	+0.8	+3.2	do
	+2.1	+2.8	+1.9	+3.4	you
	+1.0	+3.0	+2.7	+3.6	feel
	-1.2	+0.2	+3.1	+3.1	?
	-1.7	-0.5	+3.1	+3.1	Not
	-2.0	-1.4	+2.7	+2.7	.
+0.2036					

Figure 2: Sentence pair with similarity scores produced by our model when trained over PU examples (right) and over PURI examples (left). Aggregation scores (Equation 2) are shown next to words. Matrices contain alignment scores. Sentence similarities (Equation 1) shown below matrices.

Figure 2 illustrates the output of our network when trained using PU examples (right) and PURI examples (left). The former (right) fails to predict some word divergences, most likely because in training it never saw sentences mixing divergent and non-divergent words in the same example.

Table 3 shows BLEU results for our neural MT engine trained over different data configurations: The entire[5] data (all); Most similar pairs after optimising equation 4 (semb); Most similar pairs after optimising equation 3 (wemb); Finally, we apply the correction algorithm detailed in Section 3.2 (wemb+fix). Columns Ref and Fix indicate the number of original and corrected sentences (in millions) considered to train NMT.

Data	Ref (M)	Fix (M)	Test (BLEU)
OpenSubtitles English-French			
all	27.2	-	42.18
semb	18.0	-	41.95
wemb	15.5	-	43.12 (+0.94)
wemb+fix	15.5	2.5	44.19
wemb	18.0	-	43.19 (+1.01)
Paracrawl English-German			
all	22.2	-	19.27
semb	15.0	-	20.23
wemb	15.0	-	21.52 (+2.25)
wemb	17.5	-	21.97
wemb+fix	15	2.5	22.42

Table 3: BLEU scores obtained by neural MT using different subsets of the OpenSubtitles and Paracrawl corpora.

Results for wemb clearly outperform the baseline all (1^{st} rows) by about $+1.0$ and $+2.25$ BLEU (3^{rd} rows). In contrast, semb does not improve over the baseline (2^{nd} rows). Regarding OpenSubtitles, an additional gain of $+X.X$ is found when fixing 2.5 million sentences (4^{th} row). The same 2.5 million sentences did not show any improvement when added in their original form (5^{th} row).

At this moment experiments showing correction results regarding Paracrawl are not yet finished.

6 Conclusions

We presented an unsupervised method based on deep neural networks for detecting translation divergences in parallel sentence pairs. Our model also predicts misaligned words that can then be filtered out allowing for reusing some divergent sentences. We evaluated our model on two neural machine translation tasks, showing improvements when compared to training over the entire data set.

[5] The original Paracrawl corpus contains more than 100 million sentences. We reduced its size to 22.2 millions using standard filtering techniques.

7 Further Work

We plan to use our model to predict sentence embeddings over monolingual corpora allowing to collect parallel pairs through vector similarity measures.

References

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval@NAACL-HLT*, pages 497–511. The Association for Computer Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214. Association for Computational Linguistics.

Marine Carpuat, Yogarshi Vyas, and Xing Niu. 2017. Detecting cross-lingual semantic divergence for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 69–79. Association for Computational Linguistics.

Boxing Chen, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. 2016. Bilingual methods for adaptive training data selection for machine translation. In *Proceedings of the 12th biennial conference of the Association for the Machine Translation in Americas (AMTA2016)*.

Mara Chinea-Rios, Álvaro Peris, and Francisco Casacuberta. 2017. Adapting neural machine translation with parallel synthetic data. In *Proceedings of the Second Conference on Machine Translation*, pages 138–147. Association for Computational Linguistics.

Christian Federmann and Will Lewis. 2016. Microsoft speech language translation (mslt) corpus: The iwslt 2016 release for english, french and german. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT2016)*.

Pascale Fung and Percy Cheung. 2004. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Cyril Goutte, marine carpuat, and Georges Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *The Tenth Biennial Conference of the Association for Machine Translation in the Americas*.

Francis Grégoire and Philippe Langlais. 2017. Bucc 2017 shared task: a first attempt toward a deep learning framework for identifying parallel sentences in comparable corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 46–50. Association for Computational Linguistics.

Jeenu Grover and Pabitra Mitra. 2017. Bilingual word embeddings with bucketed cnn for parallel sentence extraction. In *Proceedings of ACL 2017, Student Research Workshop*, pages 11–16. Association for Computational Linguistics.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948. Association for Computational Linguistics.

Joël Legrand, Michael Auli, and Ronan Collobert. 2016. Neural network-based word alignment through score aggregation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 66–73. Association for Computational Linguistics.

Junyi Jessy Li, Marine Carpuat, and Ani Nenkova. 2014. Cross-lingual discourse relation analysis: A corpus study and a semi-supervised classification system. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 577–587. Dublin City University and Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *10th International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2786–2792. AAAI Press.

Dragos S. Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4).

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1310–III–1318. JMLR.org.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2011. *Bitext Alignment*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Yogarshi Vyas, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. *CoRR*, abs/1803.11112.

Yong Xu and François Yvon. 2016. Novel elicitation and annotation schemes for sentential and sub-sentential alignments of bitexts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Bing Zhao and Stephan Vogel. 2002. Adaptive parallel sentences mining from web bilingual news collection. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, Washington, DC, USA. IEEE Computer Society.