Context-Dependent Translation Selection Using Convolutional Neural Network

Baotian Hu[‡] **Zhaopeng Tu**[†*] **Zhengdong Lu**[†] **Hang Li**[†] **Qingcai Chen**[‡]

[‡]Intelligent Computing Research Center, Harbin Institute of Technology Shenzhen Graduate School
baotianchina@gmail.com
qingcai.chen@hitsz.edu.cn

[†]Noah's Ark Lab Huawei Technologies Co. Ltd.
tu.zhaopeng@huawei.com
lu.zhengdong@huawei.com
hangli.hl@huawei.com

Abstract

We propose a novel method for translation selection in statistical machine translation, in which a convolutional neural network is employed to judge the similarity between a phrase pair in two languages. The specifically designed convolutional architecture encodes not only the semantic similarity of the translation pair, but also the context containing the phrase in the source language. Therefore, our approach is able to capture *context-dependent* semantic similarities of translation pairs. We adopt a curriculum learning strategy to train the model: we classify the training examples into easy, medium, and difficult categories, and gradually build the ability of representing phrases and sentence-level contexts by using training examples from easy to difficult. Experimental results show that our approach significantly outperforms the baseline system by up to 1.4 BLEU points.

1 Introduction

Conventional statistical machine translation (SMT) systems extract and estimate translation pairs based on their surface forms (Koehn et al., 2003), which often fail to capture translation pairs which are grammatically and semantically similar. To alleviate the above problems, several researchers have proposed learning and utilizing semantically similar translation pairs in a continuous space (Gao et al., 2014; Zhang et al., 2014; Cho et al., 2014). The core idea is that the two phrases in a translation pair should share the same semantic meaning and have similar (close) feature vectors in the continuous space.

* Corresponding author

The above methods, however, *neglect the information of local contexts*, which has been proven to be useful for disambiguating translation candidates during decoding (He et al., 2008; Marton and Resnik, 2008). The matching scores of translation pairs are treated the same, even they are in different contexts. Accordingly, the methods fail to adapt to local contexts and lead to precision issues for specific sentences in different contexts.

To capture useful context information, we propose a convolutional neural network architecture to measure context-dependent semantic similarities between phrase pairs in two languages. For each phrase pair, we use the sentence containing the phrase in source language as the context. With the convolutional neural network, we summarize the information of a phrase pair and its context, and further compute the pair's matching score with a multi-layer perceptron. We discriminately train the model using a curriculum learning strategy. We classify the training examples according to the difficulty level of distinguishing the positive candidate from the negative candidate. Then we train the model to learn the semantic information from *easy* (basic semantic similarities) to *difficult* (context-dependent semantic similarities).

Experimental results on a large-scale translation task show that the context-dependent convolutional matching (CDCM) model improves the performance by up to 1.4 BLEU points over a strong phrase-based SMT system. Moreover, the CDCM model significantly outperforms its context-independent counterpart, proving that it is necessary to incorporate local contexts into SMT.

Contributions. Our key contributions include:

- we introduce a novel CDCM model to capture context-dependent semantic similarities between phrase pairs (Section 2);
- we develop a novel learning algorithm to train the CDCM model using a curriculum learning strategy (Section 3).

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 536–541, Beijing, China, July 26-31, 2015. ©2015 Association for Computational Linguistics

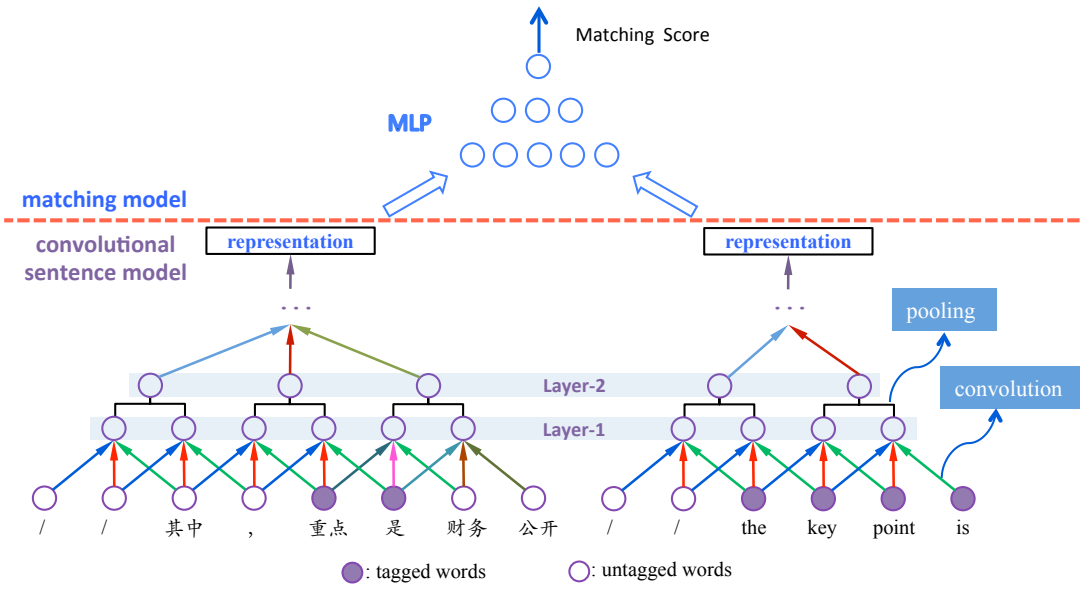

Figure 1: Architecture of the CDCM model. The *convolutional sentence model* (bottom) summarizes the meaning of the tagged sentence and target phrase, and the *matching model* (top) compares the representations using a multi-layer perceptron. "/" indicates all-zero padding turned off by the gating function.

2 Context-Dependent Convolutional Matching Model

The model architecture, shown in Figure 1, is a variant of the convolutional architecture of Hu et al. (2014). It consists of two components:

- *convolutional sentence model* that summarizes the meaning of the source sentence and the target phrase;
- *matching model* that compares the two representations with a multi-layer perceptron (Bengio, 2009).

Let $\hat{e}$ be a target phrase and $\mathbf{f}$ be the source sentence that contains the source phrase aligning to $\hat{e}$. We first project $\mathbf{f}$ and $\hat{e}$ into feature vectors $\mathbf{x}$ and $\mathbf{y}$ via the convolutional sentence model, and then compute the matching score $s(\mathbf{x}, \mathbf{y})$ by the matching model. Finally, the score is introduced into a conventional SMT system as an additional feature.

Convolutional sentence model. As shown in Figure 1, the model takes as input the embeddings of words (trained beforehand elsewhere) in $\mathbf{f}$ and $\hat{e}$. It then iteratively summarizes the meaning of the input through layers of convolution and pooling, until reaching a fixed length vectorial representation in the final layer.

In Layer-1, the convolution layer takes sliding windows on $\mathbf{f}$ and $\hat{e}$ respectively, and models all the possible compositions of neighbouring words. The convolution involves a *filter* to produce a new feature for each possible composition. Given a k-sized sliding window i on $\mathbf{f}$ or $\hat{e}$, for example, the jth convolution unit of the composition of the words is generated by:

$$\mathbf{c}_i^{(1,j)} = g(\hat{\mathbf{c}}_i^{(0)}) \cdot \phi(\mathbf{w}^{(1,j)} \cdot \hat{\mathbf{c}}_i^{(0)} + \mathbf{b}^{(1,j)}) \quad (1)$$

where

- $g(\cdot)$ is the gate function that determines whether to activate $\phi(\cdot)$;
- $\phi(\cdot)$ is a non-linear activation function. In this work, we use ReLu (Dahl et al., 2013) as the activation function;
- $\mathbf{w}^{(1,j)}$ is the parameters for the jth convolution unit on Layer-1, with matrix $\mathbf{W}^{(1)} = [\mathbf{w}^{(1,1)}, \ldots, \mathbf{w}^{(1,J)}]$;
- $\hat{\mathbf{c}}_i^{(0)}$ is a vector constructed by concatenating word vectors in the k-sized sliding widow i;
- $\mathbf{b}^{(1,j)}$ is a bias term, with vector $\mathbf{B}^{(1)} = [\mathbf{b}^{(1,1)}, \ldots, \mathbf{b}^{(1,J)}]$.

To distinguish the phrase pair from its context, we use one additional dimension in word embeddings: 1 for words in the phrase pair and 0 for the others. After transforming words to

their tagged embeddings, the convolutional sentence model takes multiple choices of composition using sliding windows in the convolution layer. Note that sliding windows are allowed to cross the boundary of the source phrase to exploit both phrasal and contextual information.

In Layer-2, we apply a local max-pooling in non-overlapping 1×2 windows for every convolution unit

$$\mathbf{c}_i^{(2,j)} = \max\{\mathbf{c}_{2i}^{(1,j)}, \mathbf{c}_{2i+1}^{(1,j)}\} \quad (2)$$

In Layer-3, we perform convolution on output from Layer-2:

$$\mathbf{c}_i{}^{(3,j)} = g(\hat{\mathbf{c}}_i{}^{(2)}) \cdot \phi(\mathbf{w}^{(3,j)} \cdot \hat{\mathbf{c}}_i^{(2)} + \mathbf{b}^{(3,j)}) \quad (3)$$

After more convolution and max-pooling operations, we obtain two feature vectors for the source sentence and the target phrase, respectively.

Matching model. The matching score of a source sentence and a target phrase can be measured as the similarity between their feature vectors. Specifically, we use the multi-layer perceptron (MLP), a nonlinear function for similarity, to compute their matching score. First we use one layer to combine their feature vectors to get a hidden state h_c:

$$h_c = \phi(w_c \cdot [\mathbf{x}_{\bar{f}_i} : \mathbf{y}_{\bar{e}_j}] + b_c) \quad (4)$$

Then we get the matching score from the MLP:

$$s(\mathbf{x}, \mathbf{y}) = MLP(h_c) \quad (5)$$

3 Training

We employ a discriminative training strategy with a max-margin objective. Suppose we are given the following triples $(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-)$ from the oracle, where $\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-$ are the feature vectors for $\mathbf{f}, \hat{e}^+, \hat{e}^-$ respectively. We have the ranking-based loss as objective:

$$L_\Theta(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) = \max(0, 1 + s(\mathbf{x}, \mathbf{y}^-) - s(\mathbf{x}, \mathbf{y}^+)) \quad (6)$$

where $s(\mathbf{x}, \mathbf{y})$ is the matching score function defined in Eq. 5, Θ consists of parameters for both the convolutional sentence model and MLP. The model is trained by minimizing the above objective, to encourage the model to assign higher matching scores to positive examples and to assign lower scores to negative examples. We use stochastic gradient descent (SGD) to optimize the model parameters Θ. We train the CDCM model with a curriculum strategy to learn the context-dependent semantic similarity at the phrase level from *easy* (basic semantic similarities between the source and target phrase pair) to *difficult* (context-dependent semantic similarities for the same source phrase in varying contexts).

3.1 Curriculum Training

Curriculum learning, first proposed by Bengio et al. (2009) in machine learning, refers to a sequence of training strategies that start small, learn easier aspects of the task, and then gradually increase the difficulty level. It has been shown that the curriculum learning can benefit the non-convex training by giving rise to improved generalization and faster convergence. The key point is that the training examples are not randomly presented but organized in a meaningful order which illustrates gradually more concepts, and gradually more complex ones.

For each positive example $(\mathbf{f}, \hat{e}^+)$, we have three types of negative examples according to the difficulty level of distinguishing the positive example from them:

- *Easy*: target phrases randomly chosen from the phrase table;
- *Medium*: target phrases extracted from the aligned target sentence for other non-overlap source phrases in the source sentence;
- *Difficult*: target phrases extracted from other candidates for the same source phrase.

We want the CDCM model to learn the following semantic information from easy to difficult:

- the *basic semantic similarity* between the source sentence and target phrase from the *easy* negative examples;
- the *general semantic equivalent* between the source and target phrase pair from the *medium* negative examples;
- the *context-dependent semantic similarities* for the same source phrase in varying contexts from the *difficult* negative examples.

Alg. 1 shows the curriculum training algorithm for the CDCM model. We use different portions of the overall training instances for different curriculums (lines 2-11). For example, we only use the

Algorithm 1 Curriculum training algorithm. Here $\mathcal{T}$ denotes the training examples, W the initial word embeddings, η the learning rate in SGD, n the pre-defined number, and t the number of training examples.

1: **procedure** CURRICULUM-TRAINING($\mathcal{T}, W$)
2: $N_1 \leftarrow$ *easy_negative*($\mathcal{T}$)
3: $N_2 \leftarrow$ *medium_negative*($\mathcal{T}$)
4: $N_3 \leftarrow$ *difficult_negative*($\mathcal{T}$)
5: $T \leftarrow N_1$
6: CURRICULUM(T, $n \cdot t$) ▷ *CUR. easy*
7: $T \leftarrow$ MIX([N_1, N_2])
8: CURRICULUM(T, $n \cdot t$) ▷ *CUR. medium*
9: **for** $step \leftarrow 1 \ldots n$ **do**
10: $T \leftarrow$ MIX([N_1, N_2, $N3$], step)
11: CURRICULUM(T, t) ▷ *CUR. difficult*
12: **procedure** CURRICULUM(T, K)
13: *iterate until reaching a local minima or K iterations*
14: *calculate* L_Θ *for a random instance in* T
15: $\Theta = \Theta - \eta \cdot \frac{\partial L_\Theta}{\partial \Theta}$ ▷ *update parameters*
16: $W = W - \eta \cdot 0.01 \cdot \frac{\partial L_\Theta}{\partial W}$ ▷ *update embeddings*
17: **procedure** MIX($\mathbf{N}$, $s = 0$)
18: $len \leftarrow$ length of $\mathbf{N}$
19: **if** $len < 3$ **then**
20: $T \leftarrow$ sampling with $[0.5, 0.5]$ from N
21: **else**
22: $T \leftarrow$ sampling with $[\frac{1}{s+2}, \frac{1}{s+2}, \frac{s}{s+2}]$ from N

training instances that consist of positive examples and *easy* negative examples in the *easy* curriculum (lines 5-6). For the latter curriculums, we gradually increase the difficulty level of the training instances (lines 7-12).

For each curriculum (lines 12-16), we compute the gradient of the loss objective L_Θ and learn Θ using the SGD algorithm. Note that we meanwhile update the word embeddings to better capture the semantic equivalence across languages during training. If the loss function L_Θ reaches a local minima or the iterations reach the predefined number, we terminate this curriculum.

4 Related Work

Our research builds on previous work in the field of context-dependent rule matching and bilingual phrase representations.

There is a line of work that employs local contexts over discrete representations of words or phrases. For example, He et al. (2008), Liu et al. (2008) and Marton and Resnik (2008) employed within-sentence contexts that consist of discrete words to guide rule matching. Wu et al. (2014) exploited discrete contextual features in the source sentence (e.g. words and part-of-speech tags) to learn better bilingual word embeddings for SMT. In this study, we take into account all the phrase pairs and directly compute phrasal similarities with convolutional representations of the local contexts, integrating the strengths associated with the convolutional neural networks (Collobert and Weston, 2008).

In recent years, there has also been growing interest in bilingual phrase representations that group phrases with a similar meaning across different languages. Based on that translation equivalents share the same semantic meaning, they can supervise each other to learn their semantic phrase embeddings in a continuous space (Gao et al., 2014; Zhang et al., 2014). However, these models focused on capturing semantic similarities between phrase pairs in the global contexts, and neglected the local contexts, thus ignored the useful discriminative information. Alternatively, we integrate the local contexts into our convolutional matching architecture to obtain context-dependent semantic similarities.

Meng et al. (2015) and Zhang (2015) have proposed independently to summary source sentences with convolutional neural networks. However, they both extend the neural network joint model (NNJM) of Devlin et al. (2014) to include the whole source sentence, while we focus on capturing context-dependent semantic similarities of translation pairs.

5 Experiments

5.1 Setup

We carry out our experiments on the NIST Chinese-English translation tasks. Our training data contains 1.5M sentence pairs coming from LDC dataset.[1] We train a 4-gram language model on the Xinhua portion of the GIGAWORD corpus using the SRI Language Toolkit (Stolcke, 2002) with modified Kneser-Ney Smoothing (Kneser and Ney, 1995). We use the 2002 NIST MT evaluation test data as the development data, and the 2004, 2005 NIST MT evaluation test data as the test data. We use minimum error rate training (Och, 2003) to optimize the feature weights. For evaluation, case-insensitive NIST BLEU (Papineni et al., 2002) is used to measure translation performance. We perform a significance test using the *sign-test* approach (Collins et al., 2005).

[1]The corpus includes LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

Models	MT04	MT05	All
Baseline	34.86	33.18	34.40
CICM	35.82^{α}	33.51^{α}	34.95^{α}
$CDCM_1$	35.87^{α}	33.58	35.01^{α}
$CDCM_2$	35.97^{α}	33.80^{α}	35.21^{α}
$CDCM_3$	$36.26^{\alpha\beta}$	$33.94^{\alpha\beta}$	$35.40^{\alpha\beta}$

Table 1: Evaluation of translation quality. $CDCM_k$ denotes the CDCM model trained in the kth curriculum in Alg. 1 (i.e., three levels of curriculum training), CICM denotes its context-independent counterpart, and "All" is the combined test sets. The superscripts α and β indicate statistically significant difference ($p < 0.05$) from Baseline and CICM, respectively.

For training the neural networks, we use 4 convolution layers for source sentences and 3 convolution layers for target phrases. For both of them, 4 pooling layers (pooling size is 2) are used, and all the feature maps are 100. We set the sliding window $k = 3$, and the learning rate $\eta = 0.02$. All the parameters are selected based on the development data. We train the word embeddings using a bilingual strategy similar to Yang et al. (2013), and set the dimension of the word embeddings be 50. To produce high-quality bilingual phrase pairs to train the CDCM model, we perform forced decoding on the bilingual training sentences and collect the used phrase pairs.

5.2 Evaluation of Translation Quality

We have two baseline systems:

- *Baseline*: The baseline system is an open-source system of the phrase-based model – Moses (Koehn et al., 2007) with a set of common features, including translation models, word and phrase penalties, a linear distortion model, a lexicalized reordering model, and a language model.

- CICM (context-*independent* convolutional matching) model: Following the previous works (Gao et al., 2014; Zhang et al., 2014; Cho et al., 2014), we calculate the matching degree of a phrase pair without considering any contextual information. Each unique phrase pair serves as a positive example and a randomly selected target phrase from the phrase table is the corresponding negative example. The matching score is also introduced into Baseline as an additional feature.

Table 1 summaries the results of CDCMs trained from different curriculums. No matter from which curriculum it is trained, the CDCM model significantly improves the translation quality on the overall test data (with gains of 1.0 BLEU points). The best improvement can be up to 1.4 BLEU points on MT04 with the fully trained CDCM. As expected, the translation performance is consistently increased with curriculum growing. This indicates that the CDCM model indeed captures the desirable semantic information by the curriculum learning from easy to difficult.

Comparing with its context-independent counterpart (CICM, Row 2), the CDCM model shows significant improvement on all the test data consistently. We contribute this to the incorporation of useful discriminative information embedded in the local context. In addition, the performance of CICM is comparable with that of $CDCM_1$. This is intuitive, because both of them try to capture the basic semantic similarity between the source and target phrase pair.

One of the hypotheses we tested in the course of this research was disproved. We thought it likely that the *difficult* curriculum ($CDCM_3$ that distinguishs the correct translation from other candidates for a given context) would contribute most to the improvement, since this circumstance is more consistent with the real decoding procedure. This turned out to be false, as shown in Table 1. One possible reason is that the "negative" examples (other candidates for the same source phrase) may share the same semantic meaning with the positive one, thus give a wrong guide in the supervised training. Constructing a reasonable set of negative examples that are more semantically different from the positive one is left for our future work.

6 Conclusion

In this paper, we propose a context-dependent convolutional matching model to capture semantic similarities between phrase pairs that are sensitive to contexts. Experimental results show that our approach significantly improves the translation performance and obtains improvement of 1.0 BLEU scores on the overall test data.

Integrating deep architecture into context-dependent translation selection is a promising way to improve machine translation. In the future, we will try to exploit contextual information at the target side (*e.g.,* partial translations).

Acknowledgments

This work is supported by China National 973 project 2014CB340301. Baotian Hu and Qinghai Chen are supported by National Natural Science Foundation of China 61173075 and 61473101. We thank Junhui Li, and the anonymous reviewers for their insightful comments.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML 2009*.

Yoshua Bengio. 2009. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP 2014*.

M. Collins, P. Koehn, and I. Kučerová. 2005. Clause restructuring for statistical machine translation. In *ACL 2005*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML 2008*.

George E Dahl, Tara N Sainath, and Geoffrey E Hinton. 2013. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *ICASSP 2013*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *ACL 2014*.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2014. Learning continuous phrase representations for translation modeling. In *ACL 2014*.

Zhongjun He, Qun Liu, and Shouxun Lin. 2008. Improving statistical machine translation using lexicalized rule selection. In *COLING 2008*.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS 2014*.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *ICASSP 1995*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL 2003*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL 2007*.

Qun Liu, Zhongjun He, Yang Liu, and Shouxun Lin. 2008. Maximum entropy based rule selection model for syntax-based statistical machine translation. In *EMNLP 2008*.

Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *ACL 2008*.

Fandong Meng, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, and Qun Liu. 2015. Encoding source language with convolutional neural network for machine translation. In *ACL 2015*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL 2003*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of Seventh International Conference on Spoken Language Processing*, volume 3, pages 901–904. Citeseer.

Haiyang Wu, Daxiang Dong, Xiaoguang Hu, Dianhai Yu, Wei He, Hua Wu, Haifeng Wang, and Ting Liu. 2014. Improve statistical machine translation with context-sensitive bilingual semantic embedding model. In *EMNLP 2014*.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word Alignment Modeling with Context Dependent Deep Neural Network. In *ACL 2013*.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *ACL 2014*.

Jiajun Zhang. 2015. Local translation prediction with global sentence representation. In *IJCAI 2015*.