Semantic-aware Deep Neural Attention Network for Machine Translation Detection*

Yangbin Shi[1,2], Jun Lu[2], Shuqin Gu[2], Qiang Wang[2], and Xiaolin Zheng[1]

[1] College of Computer Science and Technology, Zhejiang University
[2] Machine Intelligence Technology Lab, Alibaba Group
Hangzhou, China
shiyangbinxixi@163.com, xlzheng@zju.edu.cn

Abstract. Web crawling is an important way to collect a massive training corpus for building a high-quality machine translation system. However, a large amount of data collected comes from machine-translated texts rather than native speakers or professional translators, severely reducing the benefit of data scale. Traditional machine translation detection methods generally require human-crafted feature engineering and are difficult to distinguish the fine-grained semantic difference between real text and pseudo text from a modern neural machine translation system. To address this problem, we propose two semantic-aware models based on the deep neural network to automatically learn semantic features of text for monolingual scenarios and bilingual scenarios, respectively. Specifically, our models incorporate the global semantic from BERT and the local semantic from convolutional neural network together for monolingual detection and further explores the semantic consistency relationship for bilingual detection. The experimental results on the Chinese-English machine translation detection task show that our models achieve 83.12% F_1 in the monolingual detection and 85.53% F_1 in the bilingual detection respectively, which is better than the strong BERT baselines by 2.2-3.2%.

Keywords: Machine Translation Detection, Local & Global Semantic Representation

1 Introduction

As we all know, data-driven machine translation, including statistical machine translation (SMT) [25] and neural machine translation (NMT) [4, 23], strongly depends on the quality and quantity of the training corpora. For example, bilingual parallel pairs are used for supervision learning, and monolingual target data is available for language model[14] or data augmentation [20]. In practice, due to its low cost, data mining from subtitles and web crawling is one of the most popular ways to collect massive data for machine translation [12, 19]. However,

* Supported in part by the National Key R&D Program of China(No. 2018YFB1403001)

there are many noises in the collected data, which may mislead the model training and damage the performance of machine translation systems. In this work, we focus on the issue of *machine translation detection* (MTD) [1], which is a typical noise sourcing caused by the fact that a large amount of crawling data comes from machine-translated texts rather than native speakers or professional translators.

Most previous MTD work aims at SMT [1, 2, 3]. They design many human-crafted features and train binary statistical classifiers to identify whether a sentence comes from a SMT system. As SMT is notorious for long-distance reorder and is prone to generate the disfluent translation, these simple statistical classifiers can achieve good performance by adding some explicit linguistic features. However, the situation changes when turning to modern NMT systems. Specifically, NMT is modeled as a conditional language model, which is naturally good at generating fluent and grammatical translation [13]. Therefore, we argue that the previous coarse-grained MTD models cannot fit the NMT scenario, and it is necessary to design a fine-grained MTD model to distinguish the semantic bias between real text and machine-translated text.

To address this issue, we propose to model the deep semantic representation by neural network for both monolingual and bilingual scenarios. Specifically, aimed at monolingual sentence, we propose the **S**emantic-aware **I**nfluencing **A**ttention **N**etwork (SIAN) to capture the global and local semantic information of a sentence by combining BERT model[8] and Convolutional Neural Network (CNN) [11] together. SIAN integrates the important local semantic information into the global semantic information by adopting an influencing attention mechanism for obtaining the sufficient semantic representation of a sentence. In contrast, for the bilingual scenario, we further propose a **S**emantic **C**onsistency-aware **I**nteractive **A**ttention network (SCIA), which match the semantics of a target sentence with its corresponding source sentence to obtain the semantic consistency. In addition, the Part-of-Speech (POS) is used as the input to make the model better aware of the shallow syntactic information.

We compare our models with several baseline models (i.e., statistical classifier model and neural network-based models) on the outputs of four online NMT systems. Experimental results show that our proposed models outperform all of the baseline models by achieving an 83.12% F_1 in the monolingual scenario and an 85.53% F_1 in the bilingual scenario, respectively. To the best of our knowledge, we are the first to explore neural network-based techniques to tackle the machine translation detection task.

2 Related Work

Previous techniques for detecting machine-translated sentences are designed for SMT [1, 2, 3]. In the monolingual scenario, Arase et al.[3] designed a sentence-level classifier to distinguish the machine-translated sentences from a mixture of machine-translated and human-translated sentences. They utilized the phrase salad phenomenon and gappy-phrase features to detect if a sentence is fluent and grammatical. Aharoni et al.[1] utilized the common content-independent linguistic features for this detecting task. The features in their work were binary, denoting the presence or absence of each of a set of part-of-speech n-grams, as well as the function words. Both of their work adopted a binary statistical supervised classifier, i.e., SVM, to determine the likelihoods of machine-translated or human-translated sentences.

In the bilingual scenario, Antonova et al.[2] designed a phrase-based decoder for detecting machine-translated content in a Web-scraped parallel Russian-English corpus. By evaluating the BLEU score of translated content (by their decoder) against the target-side content, machine-translated content can be detected. Rarrick et al.[18] extract a variety of features, such as the number of tokens and character types, from sentences in both the source and target languages to capture words that are mistranslated by MT systems. With these features, the likelihood of a bilingual sentence pair being machine-translated can be determined.

The above work is designed for detecting the outputs of SMT by utilizing some explicit linguistic features and statistical supervised classifiers. We also address the problem as a binary classification task. In contrast, since the NMT has achieved significant success, we pay more attention to the implicit semantic features rather than such explicit linguistic features.

Data selection for machine translation system is a related area. These studies [5, 7, 14] aim to properly select data for training a subset sentence pairs from a large corpus, so that improve the performance of the MT system in the specific domain. Chen et al.[7] proposed a semi-supervised CNN based on bi-tokens (Bi-SSCNNs) for training machine translation systems from a large bilingual corpus. Moore et al.[14] use the language model to select domain-relate corpus. However, these methods are designed to select specific domain data. Our work utilizes the similar idea that detects the machine-translated sentences by relying on the neural networks for capturing more implicit information instead.

Another related area is the cross-lingual semantic textual similarity modeling, to which assesses the degree of two sentences in a different language is semantically equivalent to each other [6]. Shao et al.[21] use CNN to capture the semantic representation of the source and target sentences. Then a semantic difference vector between these two paired sentences is generated. While the aims of the tasks mentioned above are different from ours, we take the advantage of neural networks to obtain the semantic consistency information. We regard semantic consistency as an implicit feature for detecting the sentences with the semantic bias that were translated by the NMT system in the bilingual scenario.

3 Model Overview

This section,introduces our neural network-based methods for MTD task, including semantic-aware influencing attention network in monolingual scenario and semantic consistency-aware interactive attention network in bilingual scenario. The model architectures are shown in Figure 1 and Figure 2, respectively.

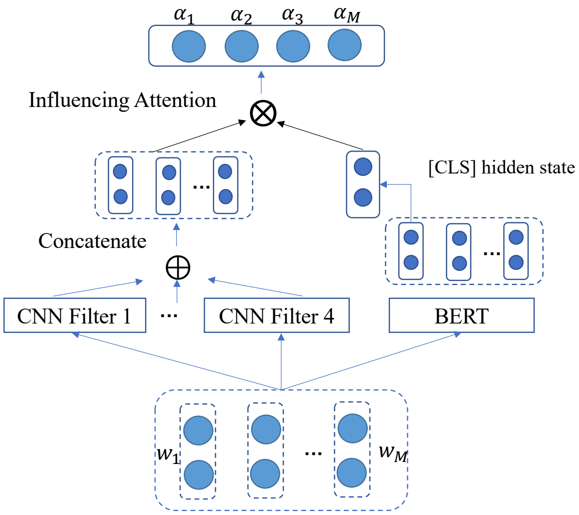


Fig. 1. Architecture of semantic-aware influencing attention network based on BERT and CNN (SIAN).

3.1 Semantic-aware Influencing Attention Network (SIAN) in Monolingual Scenario

Our problem can be formulated as follows. Given a sentence with M words, we need to judge whether the sentence is machine-translated or human-translated. We propose a semantic-aware influencing attention network (SIAN) based on BERT and CNN for this task and the model architecture is shown in Figure 1.

Global Semantic Feature Extraction by BERT Specifically, the [**CLS**] token's hidden state is used as the hidden contextual representation of a sentence.

Local Semantic Feature Extraction by CNN In order to capture the local semantic information of the sentence sufficiently, we use convolution blocks with different sizes of filters to encode the input sentence.

Let $w_i \in R^d$ be the d-dimensional word vector corresponding to the i-th word in the sentence. Let $\mathbf{X} \in R^{M\times d}$ denotes the input sentence where M is the length of the sentence with padding. A convolutional filter $\mathbf{W}_c \in R^{d\times k}$ maps k words in the respective filed to a single feature c. As we slide the filter across the whole sentence, we obtain a sequence of new features $\mathbf{c} = [c_1, c_2, ..., c_M]$.

$$c_i = f(\mathbf{X}_{i:i+k} * \mathbf{W}_c + b_c), \tag{1}$$

where $b_c \in R$ is a bias term and f is a nonlinear transformation function such as ReLU, $*$ denotes convolution operation.

SIAN Model for Machine Translation Detection We have introduced the process about one feature is extracted from one filter. Since our SIAN model utilizes multiple filters with different filter sizes to generate multiple feature maps for capturing more local n-grams semantic information of a sentence. Therefore, we obtain the final local n-grams semantic representation by concatenating the different feature maps, $\mathbf{C} = [\mathbf{c}_1; \mathbf{c}_2; ...; \mathbf{c}_n]$, where n is the number of filters.

Moreover, we capture the global semantic representation by using the $[CLS]$ token's hidden state, $\mathbf{h}_{cls}$.

Next, we utilize the global semantic vector $\mathbf{h}_{cls}$ and the convolutional features vector $\mathbf{C}$ to calculate the attention weights, which attempts to capture some important local n-grams features to supplement the global semantic information:

$$\alpha_i = \frac{exp(s(\mathbf{c}_i, \mathbf{h}_{cls}))}{\sum_{j=1}^{M} exp(s(\mathbf{c}_j, \mathbf{h}_{cls}))} \tag{2}$$

where s is a score function that calculates the importance of $\mathbf{c}_i$ in the whole n-grams semantic features. The score function is defined as:

$$s(\mathbf{c}_i, \mathbf{h}_{cls}) = tanh(\mathbf{c}_i \cdot \mathbf{W}_a \cdot \mathbf{h}_{cls}^T + b_a) \tag{3}$$

where $\mathbf{W}_a$ and b_a are weight matrix and bias respectively, tanh is a non-linear function and $\mathbf{h}_{cls}^T$ is the transpose of the $\mathbf{h}_{cls}$. Then we can get the sufficient global semantic representation $\mathbf{H}$ by integrating the import local n-grams features to global semantic vector,

$$\mathbf{H} = \mathbf{h}_{cls} + \sum_{i=1}^{M} \alpha \mathbf{c}_i \tag{4}$$

We obtain the final semantic representation $\mathbf{H}_R$ by concatenating $\mathbf{H}$ and $\mathbf{C}_{max}$ for the completeness of semantic information,

$$\mathbf{C}_{max} = max(\mathbf{C}), \mathbf{H}_R = [\mathbf{H}; \mathbf{C}_{max}] \tag{5}$$

where $\mathbf{C}_{max}$ is generated by the max-over-time pooling operation.

Later, the sequence representation $\mathbf{S}$ is obtained by using a non-linear layer:

$$\mathbf{S} = tanh(\mathbf{W}_R \mathbf{H}_R + b_R), \tag{6}$$

where $\mathbf{W}_R$ and b_R are weight matrix and bias, respectively.

We feed $\mathbf{S}$ into a linear layer, the length of whose output equals the number of class labels. Finally, we add a *softmax* layer to calculate the probability distribution for judging a sentence is machine-translated or human-translated:

$$\mathbf{y} = \mathit{softmax}(\mathbf{W}_f \mathbf{S} + b_f), \tag{7}$$

where $\mathbf{W}_f$ and b_f are the weight matrix and bias of *softmax* layer, respectively.

3.2 Semantic consistency-aware Interactive Attention Network (SCIA) in Bilingual Scenario

We further tackle this detecting task from the perspective of semantic consistency in the bilingual scenario. For instance, given a source sentence, the standard

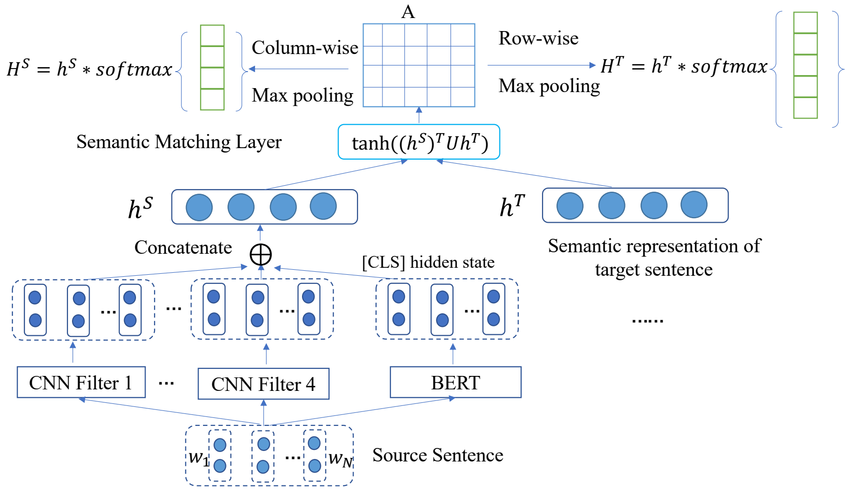


Fig. 2. Architecture of semantic consistency-aware interactive attention network (SCIA)

Table 1. An example of paired sentence.

Source	自由必须是有目标的自由，不然的话，我们便很容易感到厌倦
Human	Freedom must be **a purposeful freedom**, otherwise, we can easily get tired of it.
MT	Freedom must be ***freedom of purpose***, otherwise we will easily get bored.

human-translated sentence and machine-translated sentence of the target side are shown in Table 1.

In this example, due to the high performance of the NMT, we find that a machine-translated sentence is as fluent and grammatical as a sentence generated by human. When we focus on its semantics, we will find that its semantic information is a little different from its source sentence. Therefore, in order to better distinguish whether a sentence is machine-translated, we should further focus on whether its semantics is consistent with its corresponding source sentence.

Here, the BERT and CNN are also used to encode the global and local semantic representations in this scenario. We directly concatenate the representations of the BERT and CNN without pooling for source sentence S (similar to the target sentence T), generating the semantic vector $\mathbf{h}^S$ (semantic representation $\mathbf{h}^T$ for the target sentence). The architecture is shown in Figure 2.

SCIA model for Machine Translation Detection In the bilingual scenario, we should pay more attention to the mutual semantic relation between the source and target sentence. Thus, an interactive attention network is proposed to capture semantic consistency.

Interactive attention is an approach that enables the semantic matching layer to be aware of the current input pair, in a way that the $\mathbf{h}^S$ is able to directly influence the $\mathbf{h}^T$, and vice versa. The main idea of the interactive attention is to encourage the hidden contextual representations interactively learning semantic matching information for the source and target sentences. Then the attention weights can be calculated by applying the column-wise and row-wise max pooling over $\mathbf{A}$ matrix.

Consider the input pair (S, T) where the length of the source sentence S is N, and the length of the target sentence T is M. The matrix $\mathbf{A} \in R^{N \times M}$ can be calculated as follows:

$$\mathbf{A} = tanh((\mathbf{h}^S)^T \mathbf{U} \mathbf{h}^T + b_A), \tag{8}$$

where $\mathbf{U}$ is a weight matrix, b_A is the bias, and $(\mathbf{h}^S)^T$ denotes the transpose of the $\mathbf{h}^S$.

Later, we apply the column-wise and row-wise max pooling over the $\mathbf{A}$ matrix to generate the vectors $\mathbf{a}^s \in R^N$ and $\mathbf{a}^t \in R^M$, respectively.

$$[a^s]_i = \max_{1<n<N} [\mathbf{A}_{i,n}] \tag{9}$$

$$[a^t]_i = \max_{1<m<M} [\mathbf{A}_{m,i}] \tag{10}$$

Each element i of the vector $\mathbf{a}^t$ can be interpreted as an importance for the local n-grams semantic information around the i-th word in the representation of target sentence $\mathbf{h}^T$ according to the representation of source sentence $\mathbf{h}^S$. In the same way, each element i of the vector $\mathbf{a}^s$ can be interpreted as an importance for the local n-grams semantic information around the i-th word in the representation of source sentence $\mathbf{h}^S$ according to the representation of the target sentence $\mathbf{h}^T$.

Sequentially, we adopt the *softmax* function to the vectors $\mathbf{a}^s$ and $\mathbf{a}^t$ to generate the attention weight α and β

$$[\alpha^s]_i = \frac{exp([a^s]_i)}{\sum_{1<b<M} exp([a^s]_b)} \tag{11}$$

$$[\beta^t]_i = \frac{exp([a^t]_i)}{\sum_{1<b<N} exp([a^t]_b)} \tag{12}$$

Next, we can get the final representations of the source and target sentences, respectively:

$$\mathbf{H}^S = \mathbf{h}^S * \alpha \tag{13}$$

$$\mathbf{H}^T = \mathbf{h}^T * \beta \tag{14}$$

In addition, we apply element-wise absolute difference and element-wise dot product, which model the semantic bias information and consistency information between two semantic vectors ($\mathbf{H}^S$ and $\mathbf{H}^T$), respectively.

$$\mathbf{H}_i^{(1)} = |\mathbf{H}_i^S - \mathbf{H}_i^T| \tag{15}$$

$$\mathbf{H}_i^{(2)} = \mathbf{H}_i^S \odot \mathbf{H}_i^T \tag{16}$$

Table 2. Chinese-English data-sets

ZH-EN	**train-set**	**development-set**	**test-set**
human-trans	$3.7 * 10^6$	5000	5000
machine-trans	$3.7 * 10^6$	5000	5000

The final semantic consistent vector is got by concatenating $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$, $\mathbf{H}_F = [\mathbf{H}^{(1)}; \mathbf{H}^{(2)}]$, and then feed it to a fully connected layer to get the probability distribution for judging a sentence is machine-translated or human-translated.

4 Experiments

4.1 Data Preparation

For the purpose of evaluation, we use human-translated and machine-translated sentences to train our proposed models. For the human-translated sentences, we use the WMT18 Chinese-English (*ZH-EN*) parallel sentence pairs[3]. A few methods [9] are used to filter the *lower-quality* sentence pairs. For the machine-translated sentences, we randomly feed the source sentences (i.e., the Chinese sentences) of the above high-quality parallel corpus to four online commercial machine translators[4] for obtaining target sentences (i.e., English sentence). In this way, we can obtain large amounts of positive and negative (i.e., human- and machine-translated sentence pairs) data instances. Moreover, the source sentences are segmented and POS tagged by using an NLP toolset we developed. The target sentences are tokenized and POS tagged by using NLTK toolset[5]. The whole data-set is divided into three parts: train-set, development-set and test-set. Table 2 shows the details of the data-sets used in our experiments.

4.2 Model Parameters Settings

In our experiments, including monolingual and bilingual scenarios, the POS tags are converted to the corresponding tag embeddings, and the dimension of the word embedding and the POS tag embedding are both set to 300, all of them are randomly initialized and updated during the training process. In addition, four convolution blocks are used with kernel windows of $1, 2, 3, 4$, each with 200 feature maps. And in order to have a similar number of parameters in the CNN, the BERT model is set to be 512 hidden size and 12 layers. We use PyTorch[6] to implement our proposed models and employ the Adadelta [24] as the training algorithm, whose decay rate is set to 0.95. The regularization parameter λ is set to 10^{-4} and the initial learning rate is set to 1.0.

[3] http://www.statmt.org/wmt18/translation-task.html

[4] To our knowledge, all the four machine translators are NMT systems.

[5] https://www.nltk.org/

[6] https://pytorch.org/

Table 3. Performance of models in the monolingual scenario.

Model	**Acc**	F_1
SVM	70.93	70.84
CNN	73.31	73.79
BERT	80.01	79.89
CNN+POS	74.78	74.31
SN	81.76	81.54
SIAN	82.45	82.56
SIAN+POS	**83.01**	**83.12**

4.3 Evaluation Metric

To evaluate our models, we adopt the *Accuracy* (*Acc*) and F_1 score as metrics, where $Acc = \frac{numberofcorrectpredictions}{Totalnumberofpredictions}$ and $\mathrm{F}_1 = \frac{2*precision*Recall}{precision+Recall}$.

4.4 Model Comparison and Analysis in Monolingual Scenario

In order to evaluate the performance of our SIAN model, we compare it with the statistic classifier, i.e., SVM, and the CNN/BERT models used in data selection.

SVM: Using the common content-independent linguistic features, such as N-grams, function words and POS tags, and adopt the SVM-SMO as a classifier for this detecting task [1].

CNN/BERT: Using CNN or BERT as a sentence encoder, and then stack two fully connected layers.

SN: Semantic-aware network (SN) model is designed by us in this work, which also adopts CNN and BERT to encode the local and global semantic information of a sentence. The only difference between SN and SIAN is that SN does not utilize the influencing attention mechanism.

Table 3 shows the performance of our SIAN model and other methods. It is obvious that our SIAN model with POS tags achieves the best performance among all methods. We can find that SVM gets the worst performance because this method mainly depends on some linguistic features or rules to judge the fluency degree of a sentence for detecting the outputs of the SMT. The quality of translations in NMT has been improved significantly over SMT, and the fluency of NMT generated sentences are close to the human-translated sentences. Thus machine-translated sentences cannot be effectively identified if only rely on such features and classifiers.

Furthermore, when we compare the SN model with CNN and BERT models, we find that SN model achieves better performance. Because if only adopt CNN or BERT as a sentence encoder, which may neglect the global semantics or local semantics of a sentence, while SN model simultaneously takes into account this semantic information instead. Therefore, according to these three experimental results, we can demonstrate that it is important to combine the local and global semantic features in this task.

Table 4. Performance of models in the bilingual scenario.

Model	**Acc**	F_1
CNN	78.59	76.32
BERT	83.24	83.31
CNN-Pair	80.19	78.90
CNN-Pair+POS	80.62	79.36
SCN	84.12	84.32
SCIA	84.98	84.87
SCIA+POS	**85.35**	**85.53**

As for the SIAN model, it outperforms the SN model. Since it pays more attention to some important local n-grams semantic information that is achieved by the influencing attention mechanism. Besides, SIAN integrates the local semantic information into the global semantic information to obtain the sufficient semantic representation of a sentence.

Here, we further employ the shallow syntactic information (i.e., POS) of the sentences as an auxiliary feature to improve the performance of this task. From Table 3, we can find that models with the POS tags perform better than their corresponding models without POS tags.

4.5 Model Comparison and Analysis in the Bilingual Scenario

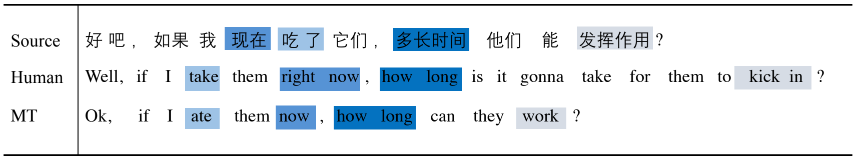
In this subsection, we compare the SCIA model with the following models.

CNN/BERT: CNN or BERT are used to encode source and target sentences. Then the encoding vectors of the sentence pair are concatenated to generate the final representation. Finally, we apply two fully connected layers to compute a unique score for a bilingual sentence pair [16].

CNN-Pair: Using CNN to capture the semantic vectors of the source and target sentences, respectively. Then generates a semantic difference vector between a sentence pair by concatenating their element-wise absolute difference and the element-wise multiplication of their semantic vectors. Finally, the feed-forward layer is used to obtain a similarity score [21].

SCN: Semantic consistency-aware network (SCN) model is designed by us in this scenario, whose architecture is similar to the SCIA model. The only difference between these two models is that the SCN model does not utilize the interactive attention mechanism.

From Table 4, it is obvious that our SCIA model with POS tags achieves the best performance. We can find that the CNN-Pair model performs better than CNN model because both CNN encode the representations of the source and target sentences without considering the semantic bias of the paired sentences. Instead, the CNN-Pair model takes advantage of the element-wise absolute difference and the element-wise multiplication of the corresponding paired sentence level embedding. It can model the relation of the source and target sentence and is conducive to identify machine-translated sentences. Although CNN-Pair model



Source	好 吧 , 如果 我 现在 吃 了 它们 , 多长时间 他们 能 发挥作用 ?
Human	Well, if I take them right now , how long is it gonna take for them to kick in ?
MT	Ok, if I ate them now , how long can they work ?

Fig. 3. A real case from our test set.

considers the relationship between the sentence pairs, it only captures the local semantic information of the source and target sentences while without taking the global semantic information into account. Thus CNN-Pair model performs less competitively than our SCN model.

As for the SCIA model, it outperforms the SCN model since the SCIA realizes the importance of the mutual relationship between a source and target sentence pair by utilizing the interactive attention mechanism. It enables the semantic matching layer to be aware of the current input pair in a way that the current semantic representation of the source sentence can directly influence the semantic representation of the target sentence and vice versa. Thus, the SCIA model can learn more semantic consistency information than the SCN model. Similar to the monolingual scenario, the POS tags bring further improvement to the CNN-Pair or SCIA model.

4.6 Case Study

Particularly, to have an intuitive understanding of our proposed model, we give a sample instance to illustrate the characteristics of the SCIA model better as shown in Figure 3. The same color corresponds to the word alignment translation. From this case, we can find that although the machine-translated sentence can be translated accurately in the word alignment level, its semantics of the whole sentence is ambiguous according to its corresponding source sentence, i.e., there is some semantic bias compared with the corresponding source sentence. Thus, the statistical classifiers tend to identify these sentences as human-translated while the SCIA model does not.

4.7 Evaluation on Neural Machine Translation Systems

We further test our SCIA model on an NMT system [4, 22].

The experiments are carried out with an open-source system called Marian [10], which is a transformer-based NMT training system[23]. We carry out experiments on the Chinese-English dataset of WMT2017 task[7]. We select 400000 sentence pairs from these datasets as the original training dataset; the development set is WMT2017's test set, which contains 2002 sentence pairs. The test set comes from WMT2018 news translation task, which contains 3981 sentence

[7] http://www.statmt.org/wmt17/translation-task.html

Table 5. BLEU scores of the WMT17 Chinese-English Translation.

Data Size	**Data Description**	**BLEU**
0.4M	Original Dataset	16.3
0.4M	Noisy Dataset	15.4
0.34M	Clean-up Dataset	15.9

pairs. Then we randomly select 30% sentences from the training data and obtain the corresponding machine-translated target sentences by four online machine translators, obtaining noisy dataset. Next, we use our proposed SCIA model to filter out the machine-translated sentence pairs from the noisy dataset, obtaining the clean-up dataset.

Table 5 shows the BLEU[15] scores of the NMT systems based on different training data. From this table, we can see that when we introduce the noise to the original data, we lost 0.9 BLEU score. Then, if we apply our SCIA model to the noisy data, the BLEU score improves the performance to 15.9 on the clean-up data, which demonstrates that the SCIA model can screen out the machine-translated sentences for improving the performance of the NMT system.

The Back-Translation method [17, 20] has been widely used in building NMT systems. Our models may improve the performance of Back-Translation further by filtering low-quality back translated sentence pairs.

5 Conclusion

In this paper, we propose two neural network models for detecting the sentences generated by NMT in monolingual and bilingual scenarios, including a semantic-aware influencing attention network (SIAN), which is used to capture important local semantic information; and a semantic consistency-aware interactive attention network (SCIA), which is used to capture semantic matching between a source and target sentence pair. Results show that our models outperform all of the baseline models by achieving an 83.12% F_1 in the monolingual scenario and an 85.53% F_1 in the bilingual scenario respectively,which is better than the strong BERT baselines by 2.2-3.2%. To the best of our knowledge, SIAN and SCIA are the first neural network-based models that are proposed to apply on the NMT output detection task.

References

[1] Aharoni, R., Koppel, M., Goldberg, Y.: Automatic detection of machine translated text and translation quality estimation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 289–295 (2014)

[2] Antonova, A., Misyurev, A.: Building a web-based parallel corpus and filtering out machine-translated text. In: Proceedings of the 4th Workshop

on Building and Using Comparable Corpora: Comparable Corpora and the Web. pp. 136–144. Association for Computational Linguistics (2011)

[3] Arase, Y., Zhou, M.: Machine translation detection from monolingual webtext. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1597–1607 (2013)

[4] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

[5] Biçici, E., Yuret, D.: Instance selection for machine translation using feature decay algorithms. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. pp. 272–283. Association for Computational Linguistics (2011)

[6] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. arXiv preprint arXiv:1708.00055 (2017)

[7] Chen, B., Kuhn, R., Foster, G., Cherry, C., Huang, F.: Bilingual methods for adaptive training data selection for machine translation. In: Proc. of AMTA. pp. 93–103 (2016)

[8] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[9] Eetemadi, S., Lewis, W., Toutanova, K., Radha, H.: Survey of data-selection methods in statistical machine translation. Machine Translation **29**(3-4), 189–223 (2015)

[10] Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Fikri Aji, A., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast neural machine translation in C++. In: Proceedings of ACL 2018, System Demonstrations. pp. 116–121. Association for Computational Linguistics, Melbourne, Australia (July 2018), http://www.aclweb.org/anthology/P18-4020

[11] Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)

[12] Lison, P., Tiedemann, J.: Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles (2016)

[13] Ma, M., Nirschl, M., Biadsy, F., Kumar, S.: Approaches for neural-network language model adaptation. Proc. Interspeech, Stockholm, Sweden pp. 259–263 (2017)

[14] Moore, R.C., Lewis, W.: Intelligent selection of language model training data (2010)

[15] Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: A method for automatic evaluation of machine translation. In: Proceedings of the ACL. pp. 311–318 (2002). https://doi.org/10.3115/1073083.1073135, http://dx.doi.org/10.3115/1073083.1073135

[16] Peris, Á., Chinea-Ríos, M., Casacuberta, F.: Neural networks classifier for data selection in statistical machine translation. The Prague Bulletin of Mathematical Linguistics **108**(1), 283–294 (2017)

[17] Poncelas, A., Shterionov, D., Way, A., Wenniger, G.M.d.B., Passban, P.: Investigating backtranslation in neural machine translation. arXiv preprint arXiv:1804.06189 (2018)

[18] Rarrick, S., Quirk, C., Lewis, W.: Mt detection in web-scraped parallel corpora. Proceedings of the Machine Translation Summit (MT Summit XIII) (2011)

[19] Resnik, P., Smith, N.A.: The web as a parallel corpus. Computational Linguistics **29**(3), 349–380 (2003)

[20] Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709 (2015)

[21] Shao, Y.: Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 130–133 (2017)

[22] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)

[23] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017)

[24] Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)

[25] Zens, R., Och, F.J., Ney, H.: Phrase-based statistical machine translation. In: Annual Conference on Artificial Intelligence. pp. 18–32. Springer (2002)