Knowledge-Based Semantic Embedding for Machine Translation

Chen Shi[†*] **Shujie Liu**[‡] **Shuo Ren**[‡] **Shi Feng**[§]
Mu Li[‡] **Ming Zhou**[‡] **Xu Sun**[†] **Houfeng Wang**[†¶]
[†]MOE Key Lab of Computational Linguistics, Peking University
[‡]Microsoft Research Asia [§]Shanghai Jiao Tong University
[¶]Collaborative Innovation Center for Language Ability
{shichen, xusun, wanghf}@pku.edu.cn sjtufs@gmail.com
{shujliu, v-shuren, muli, mingzhou}@microsoft.com

Abstract

In this paper, with the help of knowledge base, we build and formulate a semantic space to connect the source and target languages, and apply it to the sequence-to-sequence framework to propose a Knowledge-Based Semantic Embedding (KBSE) method. In our KBSE method, the source sentence is firstly mapped into a knowledge based semantic space, and the target sentence is generated using a recurrent neural network with the internal meaning preserved. Experiments are conducted on two translation tasks, the electric business data and movie data, and the results show that our proposed method can achieve outstanding performance, compared with both the traditional SMT methods and the existing encoder-decoder models.

1 Introduction

Deep neural network based machine translation, such as sequence-to-sequence (S2S) model (Cho et al., 2014; Sutskever et al., 2014), try to learn translation relation in a continuous vector space. As shown in Figure 1, the S2S framework contains two parts: an encoder and a decoder. To compress a variable-length source sentence into a fixed-size vector, with a recurrent neural network (RNN), an encoder reads words one by one and generates a sequence of hidden vectors. By reading all the source words, the final hidden vector should contain the information of source sentence, and it is called the context vector. Based on the context vector, another RNN-based neural network is used to generate the target sentence.

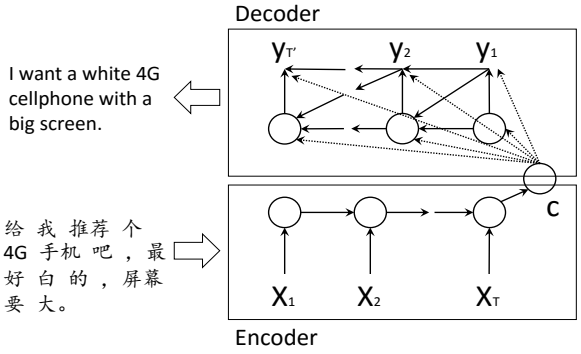

Figure 1: An illustration of the RNN-based neural network model for Chinese-to-English machine translation

The context vector plays a key role in the connection of source and target language spaces, and it should contain all the internal meaning extracted from source sentence, based on which, the decoder can generate the target sentence keeping the meaning unchanged. To extract the internal meaning and generate the target sentence, S2S framework usually needs large number of parameters, and a big bilingual corpus is acquired to train them.

In many cases, the internal meaning is not easy to learn, especially when the language is informal. For the same intention, there are various expressions with very different surface string, which aggravates the difficulty of internal meaning extraction. As shown in Table 1, there are three different expressions for a same intention, a customer wants a white 4G cellphone with a big screen. The first and second expressions (Source1 and Source2) are wordy and contain lots of verbiage. To extract the internal meaning, the encoder should ignore these verbiage and focus on key information. This is hard for the encoder-decoder mechanism, since it is not defined or formulated that what kind of information is key information. The meaning s-

This work was done while the first author was visiting Microsoft Research.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 2245–2254,
Berlin, Germany, August 7-12, 2016.

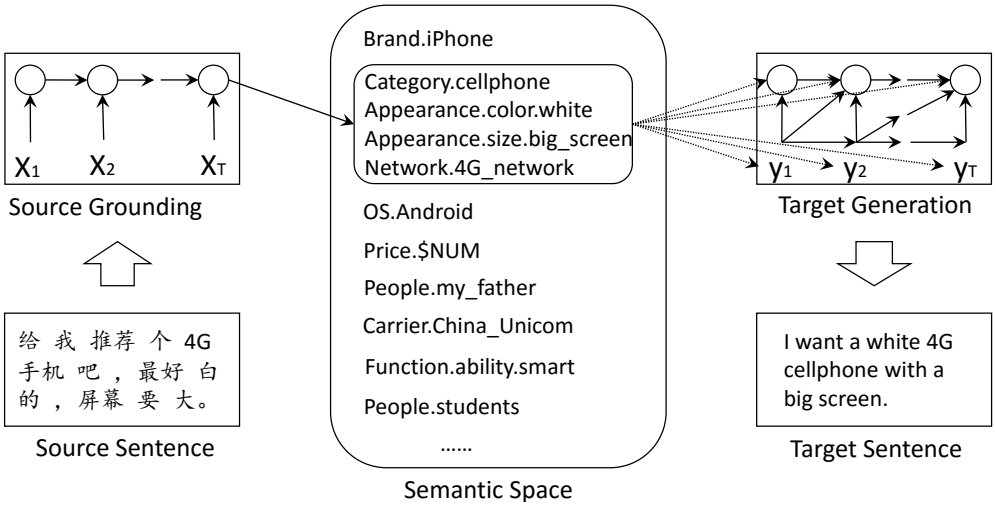

Figure 2: An illustration of Knowledge-Based Semantic Embedding (KBSE).

Source1	啊， 那个 有 大 屏幕 的 4G 手机 吗？ 要 白色 的 。
Source2	给 我 推荐 个 4G 手机 吧， 最好 白 的， 屏幕 要 大 。
Source3	我 想 买个 白色 的 大 屏幕 的 4G 手机 。
Intention	I want a white 4G cellphone with a big screen.
Enc-Dec	I need a 4G cellphone with a big screen.

Table 1: An example of various expressions for a same intention.

pace of the context vector is only a vector space of continuous numbers, and users cannot add external knowledge to constrain the internal meaning space. Therefore, the encoder-decoder system (Enc-Dec) does not generate the translation of "白色的"/"white", and fails to preserve the correct meaning of Source1, shown in Table 1.

No matter how different between the surface strings, the key information is the same (*want, white, 4G, big screen, cellphone*). This phenomenon motivates a translation process as: we firstly extract key information (such as entities and their relations) from the source sentence; then based on that, we generate target sentence, in which entities are translated with unchanged predication relations. To achieve this, background knowledge (such as, phone/computer, black/white, 3G/4G) should be considered.

In this paper, we propose a Knowledge-Based Semantic Embedding (KBSE) method for machine translation, as shown in Figure 2. Our KBSE contains two parts: a **Source Grounding** part to extract semantic information in source sentence, and a **Target Generation** part to generate target sentence. In KBSE, source monolingual data and a knowledge base is leveraged to learn an explicit semantic vector, in which the grounding space is defined by the given knowledge base, then the same knowledge base and a target monolingual data are used to learn a natural language generator, which produce the target sentence based on the learned explicit semantic vector. Different from S2S models using large bilingual corpus, our KBSE only needs monolingual data and corresponding knowledge base. Also the context/semantic vector in our KBSE is no longer implicit continuous number vector, but explicit semantic vector. The semantic space is defined by knowledge base, thus key information can be extracted and grounded from source sentence. In such a way, users can easily add external knowledge to guide the model to generate correct translation results.

We conduct experiments to evaluate our KBSE on two Chinese-to-English translation tasks, one in electric business domain, and the other in movie domain. Our method is compared with phrasal SMT method and the encoder-decoder method, and achieves significant improvement in both BLEU and human evaluation. KBSE is also combined with encoder-decoder method to get further improvement.

In the following, we first introduce our framework of KBSE in section 2, in which the details of **Source Grounding** and **Target Generation** are illustrated. Experiments is conducted in Section 3. Discussion and related work are detailed in Section 4, followed by conclusion and future work.

2 KBSE: Knowledge-Based Semantic Embedding

Our proposed KBSE contains two parts: **Source Grounding** part (in Section 2.1) embeds the source sentence into a knowledge semantic space, in which the grounded semantic information can be represented by semantic tuples; and **Target Generation** part (in Section 2.2) generates the target sentence based on these semantic tuples.

2.1 Source Grounding

Source	啊，那个有大屏幕的4G手机吗？要白色的。
Tuples	*Category.cellphone* *Appearance.color.white* *Appearance.size.big_screen* *Network.4G_network*

Table 2: Source sentence and the grounding result. Grounding result is organized as several tuples.

As shown in Table 2, given the source sentence, Source Grounding part tries to extract the semantic information, and map it to the tuples of knowledge base. It is worth noticing that the tuples are language-irrelevant, while the name of the entities inside can be in different languages. To get the semantic tuples, we first use RNN to encode the source sentence into a real space to get the sentence embedding, based on which, corresponding semantic tuples are generated with a neural-network-based hierarchical classifier. Since the knowledge base is organized in a tree structure, the tuples can be seen as several paths in the tree. For tuples in Table 2, Figure 3 shows the corresponding paths (in solid lines).

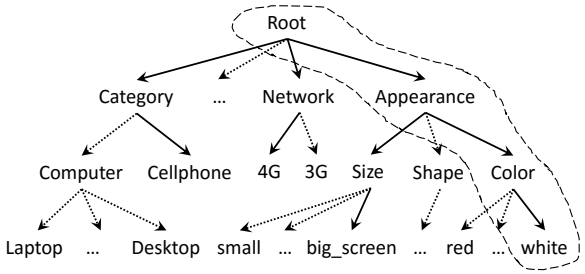

Figure 3: Illustration of the tuple tree for Table 2. Each tuple extracted from source sentence can be represented as a single path (solid line) in tuple tree. There are 4 solid line paths representing 4 tuples of Table 2. The path circled in dashed lines stands for the tuple *Appearance.color.white*.

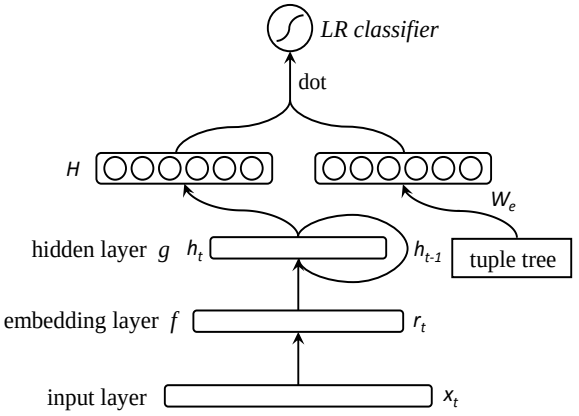

Figure 4: Illustration of Source Grounding. The input sentence **x** is transformed through an embedding layer f and a hidden layer g. Once we get the sentence embedding H, we calculate the inner product of H and the weight W_e for the specific edge e, and use a logistic regression as the classifier to decide whether this edge should be chosen.

2.1.1 Sentence Embedding

Sentence embedding is used to compress the variable-length source sentence into a fixed-size context vector. Given the input sentence $\mathbf{x} = (x_1 \ ... \ x_T)$, we feed each word one by one into an RNN, and the final hidden vector is used as the sentence embedding. In detail, as shown in Figure 4, at time-stamp t, an input word x_t is fed into the neural network. With the embedding layer f, the word is mapped into a real vector $r_t = f(x_t)$. Then the word embedding r_t is fed into an RNN g to get the hidden vector $h_t = g(r_t, h_{t-1})$. We input the words one by one at time $1, 2, ..., T$, and get the hidden vectors $h_1, h_2, ..., h_T$. The last hidden state h_T should contain all the information of the input sentence, and it is used as the sentence embedding H. To model the long dependency and memorize the information of words far from the end, Gated Recurrent Unit(GRU) (Cho et al., 2014) is leveraged as the recurrent function g.

2.1.2 Tuple Generation

In our system, we need a tuple tree for tuple generation. For those knowledge base who is naturally organized as tree structure, such as Freebase, we use its own stucture. Otherwise, we manually build the tuple tree as the representation of the introduced knowledge base. Given a knowl-

edge base for a specific domain, we divide the intention of this domain into several classes, while each class has subclasses. All the classes above can be organized as a tree structure, which is the tuple tree we used in our system, as shown in Figure 3. It is worth noticing that the knowledge base captures different intentions separately in different tree structures.

Following the hierarchical log-bilinear model (HLBL) (Mnih and Hinton, 2009; Mikolov et al., 2013), based on the sentence embedding H, we build our neural-network-based hierarchical classifier as follows: Each edge e of tuple tree has a weight vector w_e, which is randomly initialized, and learned with training data. We go through the tuple tree top-down to find the available paths. For each current node, we have a classifier to decide which children can be chosen. Since several children can be chosen at the same time independently, we use logistic regression as the classifier for each single edge, rather than a softmax classifier to choose one best child node.

For the source sentence and corresponding tuples in table 2, in the first layer, we should choose three children nodes: *Category*, *Appearance* and *Network*, and in the second layer with the parent node *Appearance*, two children nodes *color* and *size* should be selected recursively. As shown in Figure 4, the probability to choose an edge e with its connected child is computed as follows:

$$p(1|e,H) = \frac{1}{1+e^{-w_e \cdot H}} \tag{1}$$

where the operator $\cdot$ is the dot product function. The probability of the tuples conditioned on the source sentence $p(S|x_1 \ ... \ x_T)$ is the product of all the edges probabilities, calculated as follows:

$$\begin{aligned} p(S|x_1 \ ... \ x_T) &= p(S|H) \\ &= \prod_{e \in C} p(1|e,H) \prod_{e' \notin C} p(0|e',H) \end{aligned}$$

where $p(1|e,H)$ is the probability for an edge e belonging to the tuple set S, and $p(0|e',H)$ is the probability for an edge e' not in the tuple set S.

2.2 Target Generation

With the semantic tuples grounded from source sentence, in this section, we illustrate how to generate target sentence. The generation of the target sentence is another RNN, which predicts the next word y_{t+1} conditioned on the semantic vector C and all the previously predicted words $y_1, \ ..., \ y_t$. Given current word y_t, previous hidden vector h_{t-1}, and the semantic vector C, the probability of next target word y_{t+1} is calculated as:

$$h_t = g(h_{t-1}, y_t, C) \tag{2}$$

$$p(y_{t+1}|y_1 ... y_t, C) = \frac{e^{s(y_{t+1},h_t)}}{\sum_{y'} e^{s(y',h_t)}} \tag{3}$$

where equation (2) is used to generate the next hidden vector h_t, and equation (3) is the *softmax* function to compute the probability of the next word y_{t+1}. For the recurrent function g in equation (2), in order to generate target sentence preserving the semantic meaning stored in C , we modified GRU (Cho et al., 2014) following (Wen et al., 2015; Feng et al., 2016):

$$\begin{aligned} r_t &= \sigma(W^r y_t + U^r h_{t-1} + V^r c_t) \\ h_t^{'} &= \tanh(W y_t + U(r_t \odot h_{t-1}) + V c_t) \\ z_t &= \sigma(W^z y_t + U^z h_{t-1} + V^z c_t) \\ d_t &= \sigma(W^d y_t + U^d h_{t-1} + V^d c_t) \\ c_t &= d_t \odot c_{t-1} \\ h_t &= (1 \ - \ z_t) \odot h_t^{'} \ + \ z_t \odot h_{t-1} + \tanh(V^h c_t) \end{aligned}$$

in which, c_t is the semantic embedding at time t, which is initialized with C, and changed with a extraction gate d_t. The introduced extraction gate d_t retrieve and remove information from the semantic vector C to generate the corresponding target word.

To force our model to generate the target sentence keeping information contained in C unchanged, two additional terms are introduced into the cost function:

$$\sum_t \log(p(y_t|C)) + \|c_T\|_2 + \frac{1}{T}\sum_{j=1}^{T} \|d_t - d_{t-1}\|_2$$

where the first term is log-likelihood cost, the same as in the encoder-decoder. And the other two terms are introduced penalty terms. $\|c_T\|_2$ is for forcing the decoding neural network to extract as much information as possible from the semantic vector C, thus the generated target sentence keeps the same meaning with the source sentence. The third term is to restrict the extract gate from extracting too much information in semantic vector C at each time-stamp.

For the semantic tuples in Table 2, our modified RNN generates the target sentence word by word, until meets the end symbol character: "*I want a white 4G cellphone with a big screen.*".

2.3 Combination

The two components of KBSE (**Source Grounding** and **Target Generation**) are separately trained, and can be used in three ways:

- **Source Grounding** can be used to do semantic grounding for a given sentence and get the key information as a form of tuples;
- **Target Generation** can generate a natural language sentence based on the existing semantic tuples;
- Combining them, KBSE can be used to translation a source sentence into another language with a semantic space defined by a given knowledge base.

3 Experiments

To evaluate our proposed KBSE model, in this section, we conduct experiments on two Chinese-to-English translation tasks. One is from electric business domain, and the other is from movie domain.

3.1 Baseline and Comparison Systems

We select two baseline systems. The first one is an in-house implementation of hierarchical phrase-based SMT (Koehn et al., 2003; Chiang, 2007) with traditional features, which achieves a similar performance to the state-of-the-art phrase-based decoder in Moses [1] (Koehn et al., 2007). The 4-gram language model is trained with target sentences from training set plus the Gigaword corpus [2]. Our phrase-based system is trained with MERT (Och, 2003). The other system is the encoder-decoder system (van Merriënboer et al., 2015) [3], based on which our KBSE is implemented.

We also combine KBSE with encoder-decoder system, by adding the knowledge-based semantic embedding to be another context vector. Hence, for the decoder there are two context vectors, one from the encoder and the other is generated by the **Semantic Grounding** part. We call this model Enc-Dec+KBSE.

For our proposed KBSE, the number of hidden units in both parts are 300. Embedding size of both source and target are 200. Adadelta (Zeiler, 2012) is leveraged as the optimizer for neural network training. The batch size is set to 128, and learning rate is initialized as 0.5. The model weights are randomly initialized from uniform distribution between [-0.005, 0.005].

Source Sentence	Semantic Tuples
我要 iPhone 移 版 的	*Category.cellphone* *Carrier.China_Mobile* *Brand.iPhone*
黑客帝国 是 一部 由 沃卓斯基 兄弟 执导 的 科幻 电影 ， 影片 语言 为 英语 。	*Name.The_Matrix* *Genre.science_fiction* *Director.Wachowski_bro* *Language.English*
Semantic Tuples	**Target Sentence**
Category.cellphone *Appearance.color.white* *Appearance.size.big_screen* *Network.4G_network*	I want a white 4G phone with a big screen .
Name.Pirates_of_Caribbean *Released.2003* *Country.America* *Starring.Johnny_Depp*	The Pirates of the Caribbean is a 2003 American film, starring Johnny Depp .

Table 3: Illustration of dataset structure in this paper. We show one example for both corpus in both part, respectively.

3.2 Dataset Details

To train our KBSE system, we only need two kinds of pairs: the pair of source sentence and semantic tuples to train our **Source Grounding**, the pair of semantic tuples and target sentence to train our **Target Generation**. Examples of our training data in the electric business and movie domains are shown in Table 3. To control the training process of KBSE, we randomly split 1000 instances from both corpus for validation set and another 1000 instances for test set. Our corpus of electric business domain consists of bilingual sentence pairs labeled with KB tuples manually [4], which is a collection of source-KB-target triplets. For the Movie domain, all the data are mined from web, thus we only have small part of source-KB-target triplets. In order to show the advantage of our proposed KBSE, we also mined source-KB pairs and KB-target pairs separately. It should be noted that, similar as the encoder-decoder method, bilingual data is needed for Enc-Dec+KBSE, thus with the added knowledge tuples, Enc-Dec+KBSE are trained with source-KB-target triplets.

[1] http://www.statmt.org/moses/

[2] https://catalog.ldc.upenn.edu/LDC2011T07

[3] The implementation is from https://github.com/mila-udem/blocks-examples

[4] Due to the coverage problem, knowledge bases of common domain (such as Freebase) are not used in this paper.

	Electric Business			Movie		
Model	**BLEU**	**HumanEval**	**Tuple F-score**	**BLEU**	**HumanEval**	**Tuple F-score**
SMT	54.30	78.6	-	42.08	51.4	-
Enc-Dec	60.31	90.8	-	44.27	65.8	-
KBSE	62.19	97.1	92.6	47.83	72.4	80.5
Enc-Dec + KBSE	64.52	97.9	-	46.35	74.6	-
KBSE upperbound	63.28	98.2	100	49.68	77.1	100

Table 4: The BLEU scores, human evaluation accuracy, tuple F-score for the proposed KBSE model and other benchmark models.

Our electric business corpus contains **50,169** source-KB-target triplets. For this data, we divide the intention of electric business into 11 classes, which are *Category, Function, Network, People, Price, Appearance, Carrier, Others, Performance, OS* and *Brand*. Each class above also has subclasses, for example *Category* class has subclass *computer* and *cellphone*, and *computer* class can be divided into *laptop, tablet_PC, desktop* and *AIO*.

Our movie corpus contains **44,826** source-KB-target triplets, together with **76,134** source-KB pairs and **85,923** KB-target pairs. The data is crawling from English Wikipedia [5] and the parallel web page in Chinese Wikipedia [6]. Simple rule method is used to extract sentences and KB pairs by matching the information in the infobox and the sentences in the page content. Since not all the entities from Chinese wikipedia has english name, we have an extra entity translator to translate them. For a fair comparison, this entity translator are also used in other systems. Due to the whole process is semi-automatic, there may be a few irregular results within. We divided the intention of movie data into 14 classes, which are *BasedOn, Budget, Country, Director, Distributor, Genre, Language, Name, Producer, Released, Starring, Studio, Theme* and *Writer*.

3.3 Evaluation

We use BLEU (Papineni et al., 2002) as the automatical evaluation matrix, significant testing is carried out using bootstrap re-sampling method (Koehn, 2004) with a 95% confidence level. As an addition, we also do human evaluation for all the comparison systems. Since the first part **Source Grounding** of our KBSE is separately trained, the F-score of KB tuples is also evaluated. Table 4 lists evaluation results for the electric business and movie data sets.

3.3.1 BLEU Evaluation

From Table 4, we can find that our proposed method can achieve much higher BLEU than SMT system, and we can also achieve 1.9 and 3.6 BLEU points improvement compared with the raw encoder-decoder system on both eletric business and movies data.

For the Enc-Dec+KBSE method, with the same training data on electric business domain, introducing knowledge semantic information can achieve about 4 BLEU points compared with the encoder-decoder and more than 2 BLEU points compared with our KBSE. Compared with encoder-decoder, Enc-Dec+KBSE method leverages the constrained semantic space, so that key semantic information can be extracted. Compared with KBSE, which relies on the knowledge base, Enc-Dec+KBSE method can reserve the information which is not formulated in the knowledge base, and also may fix errors generated in the source grounding part.

Since Enc-Dec+KBSE can only be trained with source-KB-target triplets, for the movie dataset, the performance is not as good as our KBSE, but still achieves a gain of more than 2 BLEU point compared with the raw Enc-Dec system. On movie data, our KBSE can achieve significant improvement compared with the models (SMT, Enc-Dec, Enc-Dec+KBSE) only using bilingual data. This shows the advantage of our proposed method, which is our model can leverage monolingual data to learn **Source Grounding** and **Target Generation** separately.

We also separately evaluate the **Source Grounding** and **Target Generation** parts. We evaluate the F-score of generated KB tuples

[5] https://en.wikipedia.org
[6] https://zh.wikipedia.org

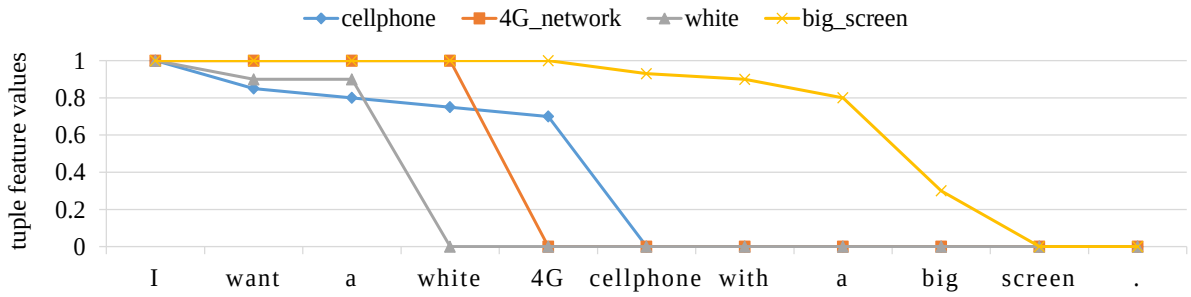

Figure 5: An example showing how the KB tuples control the tuple features flowing into the network via its learned semantic gates.

compared with the golden KB tuples. The result shows that our semantic grounding performance is quite high (92.6%), which means the first part can extract the semantic information in high coverage and accuracy. We evaluate the translation result by feeding the Target Generation network with human labeled KB tuples. The translation result (shown as KBSE upperbound in Table 4) with golden KB tuples can achieve about 1.1 and 1.8 BLEU scores improvement compared with KBSE with generated KB tuples in both dataset.

3.3.2 Human Evaluation

For the human evaluation, we do not need the whole sentence to be totally right. We focus on the key information, and if a translation is right by main information and grammar correction, we label it as correct translation, no matter how different of the translation compared with the reference on surface strings. Examples of correct and incorrect translations are shown in Table 5. As shown in Table 4, the human evaluation result shares the same trend as in BLEU evaluation. Our proposed method achieves the best results compared with SMT and raw encoder-decoder. In our method, important information are extracted and normalized by encoding the source sentence into the semantic space, and the correct translation of important information is key for human evaluation, thus our method can generate better translation.

3.4 Qualitative Analysis

In this section, we compare the translation result with baseline systems. Generally, since KB is introduced, our model is good at memorizing the key information of the source sentence. Also thanks to the strong learning ability of GRU, our model rarely make grammar mistakes. In many translations generated by traditional SMT, key information is lost. Encoder-Decoder system does much better, but some key information is also lost or even repetitively generated. Even for a long source sentence with a plenty of intentions, our model can generate the correct translation.

Target	I want a black Dell desktop.
Correct	I want a **Dell black** desktop. Could you please recommend me a black Dell desktop?
Incorrect	I want a **white** Dell desktop. I want a black Dell **laptop**. I want a black Dell desktop **desktop**.

Table 5: Some examples of which kind of sentence can be seen as a correct sentence and which will be seen as incorrect in the part of human evaluation.

To show the process of Target Generation, Figure 5 illustrates how the KB-tuples control the target sentence generation. Taking the semantic tuple *Appearance.color.white* as an example, the GRU keeps the feature value almost unchanged until the target word "white" is generated. Almost all the feature values drop from 1 to 0, when the corresponding words generated, except the tuple *Appearance.size.big_screen*. To express the meaning of this tuple, the decoding neural network should generate two words, "big" and "screen". When the sentence finished, all the feature values should be 0, with the constraint loss we introduced in Section 2.2.

Table 6 lists several translation example generated by our system, SMT system and the Encoder-Decoder system. The traditional SMT model sometimes generate same words or phrases several times, or some information is not translated. But our model rarely repeats or lose information. Besides, SMT often generate sentences unreadable, since some functional words are lost. But for KB-

Source	啊， 那个 有 大 屏幕 的 4G 手机 吗 ？ 要 白色 的 。
Reference	I want a 4G network cellphone with China Telecom supported.
KBSE	I need a white 4G cellphone with China Telecom supported.
Enc-Dec	I want a 3G cellphone with China Telecom.
SMT	Ah, that has a big screen, 4G network cellphone? give white.
Source	黑客帝国 是 一部 2003 年 由 沃卓斯基 兄弟 执导 的 电影， 里维斯 主演， 影片 语言 为 英语 。
Reference	The Matrix is a 2003 English film directed by Wachowski Brothers, starring Keanu Reeves.
KBSE	The Matrix is a 2003 English movie starring Keanu Reeves, directed by Wachowski Brothers.
Enc-Dec	The Matrix is a 2013 English movie directed by Wachowski, starring Johnny Depp.
SMT	The Matrix is directed by the Wachowski brothers film, and starring film language English.

Table 6: Examples of some translation results for our proposed KBSE system and the baseline systems.

SE, the target sentence is much easier to read. The Encoder-Decoder model learns the representation of the source sentence to a hidden vector, which is implicit and hard to tell whether the key information is kept. However KBSE learns the representation of the source sentence to a explicit tuple embedding, which contains domain specific information. So sometimes when encoder-decoder cannot memorize intention precisely, KBSE can do better.

3.5 Error Analysis

Our proposed KBSE relies on the knowledge base. To get the semantic vector of source sentence, our semantic space should be able to represent any necessary information in the sentence. For example, since our designed knowledge base do not have tuples for number of objects, some results of our KBSE generate the entities in wrong plurality form. Since our KBSE consists of two separate parts, the **Source Grounding** part and the **Target Generation** part, the errors generated in the first part cannot be corrected in the following process. As we mentioned in Section 3.3.1, combining KBSE with encoder-decoder can alleviate these two problems, by preserving information not captured and correct the errors generated in source grounding part.

4 Related Work

Unlike previous works using neural network to learn features for traditional log-linear model (Liu et al., 2013; Liu et al., 2014), Sutskever et al. (2014) introduced a general end-to-end approach based on an encoder-decoder framework. In order to compress the variable-sized source sentence into a fixed-length semantic vector, an encoder RNN reads the words in source sentence and generate a hidden state, based on which another decoder RNN is used to generate target sentence. Different from our work using a semantic space defined by knowledge base, the hidden state connecting the source and target RNNs is a vector of implicit and inexplicable real numbers.

Learning the semantic information from a sentence, which is also called semantic grounding, is widely used for question answering tasks (Liang et al., 2011; Berant et al., 2013; Bao et al., 2014; Berant and Liang, 2014). In (Yih et al., 2015), with a deep convolutional neural network (CNN), the question sentence is mapped into a query graph, based on which the answer is searched in knowledge base. In our paper, we use RNN to encode the sentence to do fair comparison with the encoder-decoder framework. We can try using CNN to replace RNN as the encoder in the future.

To generate a sentence from a semantic vector, Wen et al. (2015) proposed a LSTM-based natural language generator controlled by a semantic vector. The semantic vector memorizes what information should be generated for LSTM, and it varies along with the sentence generated. Our **Target Generation** part is similar with (Wen et al., 2015), while the semantic vector is not predefined, but generated by the **Source Grounding** part.

5 Conclusion and Future Work

In this paper, we propose a Knowledge Based Semantic Embedding method for machine translation, in which **Source Grounding** maps the source sentence into a semantic space, based on which **Target Generation** is used to generate the translation. Unlike the encoder-decoder neural network, in which the semantic space is implicit, the semantic space of KBSE is defined by a given knowledge base. Semantic vector generated by KBSE can extract and ground the key information, with the help of knowledge base, which is preserved in the translation sentence. Experiments are conducted on a electronic business and movie data sets,

and the results show that our proposed method can achieve significant improvement, compared with conventional phrase SMT system and the state-of-the-art encoder-decoder system.

In the future, we will conduct experiments on large corpus in different domains. We also want to introduce the attention method to leverage all the hidden states of the source sentence generated by recurrent neural network of **Source Grounding**.

Acknowledgement

We thank Dongdong Zhang, Junwei Bao, Zhirui Zhang, Shuangzhi Wu and Tao Ge for helpful discussions. This research was partly supported by National Natural Science Foundation of China (No.61333018 No.61370117) and Major National Social Science Fund of China (No.12&ZD227).

References

Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. 2014. Knowledge-based question answering as machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 967–976, Baltimore, Maryland, June. Association for Computational Linguistics.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1415–1425.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Shi Feng, Shujie Liu, Mu Li, and Ming Zhou. 2016. Implicit distortion and fertility models for attention-based encoder-decoder NMT model. *CoRR*, abs/1601.03317.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn, 2004. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, chapter Statistical Significance Tests for Machine Translation Evaluation.

Percy Liang, Michael I. Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *ACL*, pages 590–599.

Lemao Liu, Taro Watanabe, Eiichiro Sumita, and Tiejun Zhao. 2013. Additive neural networks for statistical machine translation. In *ACL (1)*, pages 791–801.

Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. 2014. A recursive recurrent neural network for statistical machine translation.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088.

Franz Josef Och. 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, July. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Bart van Merriënboer, Dzmitry Bahdanau, Vincent Dumoulin, Dmitriy Serdyuk, David Warde-Farley, Jan Chorowski, and Yoshua Bengio. 2015. Blocks and fuel: Frameworks for deep learning. *arXiv preprint arXiv:1506.00619*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal, September. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, July. Association for Computational Linguistics.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.