Unsupervised Automatic Text Style Transfer using LSTM

Mengqiao Han[1], Ou Wu[2] *, and Zhendong Niu[1] **

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[2] Center for Applied Mathematics, Tianjin University, Tianjin, China
michellehan@bit.edu.cn, wuou@tju.edu.cn, zniu@bit.edu.cn

Abstract. In this paper, we focus on the problem of text style transfer which is considered as a subtask of paraphrasing. Most previous paraphrasing studies have focused on the replacements of words and phrases, which depend exclusively on the availability of parallel or pseudo-parallel corpora. However, existing methods can not transfer the style of text completely or be independent from pair-wise corpora. This paper presents a novel sequence-to-sequence (Seq2Seq) based deep neural network model, using two switches with tensor product to control the style transfer in the encoding and decoding processes. Since massive parallel corpora are usually unavailable, the switches enable the model to conduct unsupervised learning, which is an initial investigation into the task of text style transfer to the best of our knowledge. The results are analyzed quantitatively and qualitatively, showing that the model can deal with paraphrasing at different text style transfer levels.

Keywords: Style Transfer, Unsupervised Learning, Seq2Seq Models

1 Introduction

Recently, style transfer has received increasing and tremendous attention among researchers from various disciplines. This technique has been successfully applied in artistic style transfer of pictures. For example, Leon A. Gatys [10] used deep neural networks to capture the style of a source picture and the semantic content of a target picture independently, and then transferred the style of the target image by combining the target picture's semantic content with the source picture's style. However, the method proposed in image style transfer cannot be readily applied to the domain of natural language processing. Research on text style transfer is still in an early stage with a wealth of topics remains to be explored.

In natural language processing, style transfer is a part of paraphrasing. Paraphrasing is to express same ideas or present same information [2] in alternative

* Prof. Ou Wu is the corresponding author.
** Prof. Zhendong Niu is the corresponding author.

ways, which is an evident subtask in many natural language processing applications, such as Information Retrieval (IR), Question Answering (QA), Information Extraction (IE), Summarization and Natural Language Generation [23].

Several studies have been dealing with paraphrase expression such as a word (lexical paraphrase) or a short phrase (phrasal paraphrase). Such paraphrasing is also called synonymizing, retaining same or similar lexical meaning of the word or the short phrase. The emergence of lexical databases such as WordNet was a significant milestone in this area.

When paraphrase expression becomes longer and more complicated, such as when a different sets of phrases are used to convey the same meaning, it is called syntactic paraphrase. For example, PPDB, a paraphrase database [9], provides an enormous collection of lexical, phrasal, and syntactic paraphrases. PPDB is released in six sizes (S to XXXL), ranging from highest precision/lowest recall to lowest average precision/highest recall. Many researchers have utilized this database to facilitate their research and have generated satisfactory results. Based on PPDB, Beltagy et al. [3] used Markov Logic Networks, and Bjerva et al. [4] adopted a system with formal semantics to recognize textual entailment (RTE). Ji and Eisenstein [14] combined latent features with fine-grained n-gram overlap features. Han et al. [11] used a lexical similarity feature that combined POS tagging, LSA word similarity and WordNet knowledge. Sultan et al. [29] considered the proportions of aligned content words in the two input sentences as semantic textual similarity scores to determine semantic text similarity. Post et al. [22] used a semi-Markov CRF for phrase-based monolingual alignment. Ganitkevitch et al. [8] introduced a method to learn syntactically-informed paraphrases for natural language generation. Yu and Dredze [33] and Rastogi et al. [24] made contributions to improve lexical embeddings. Most studies above applied supervised methods and relied heavily on parallel or pseudo-parallel corpora.

However, existing paraphrasing studies listed above have not focused on systematically transferring the style or register of a text. Systematic transformation is still a relatively unexplored field. Text style transfer focuses on generation, which is one of three related problems [23]: recognition (i.e. identifying whether two textual units are paraphrases of each other), extraction (i.e. extracting paraphrase instances from a thesaurus or a corpus) and generation (i.e. generating a reference paraphrase given a source text) [20]. Text style transfer can be ideally described as follows: given two sets of texts T_1 and T_2 with different styles S_1 and S_2 respectively, the model takes a sample $t_1 \in T_1$ in style S_1 as input and changes the text style from S_1 to S_2. The output t_new is expected to be in style S_2 but carries the same meaning as t_1. The task is useful because it has many potential applications, such as mimicking celebrities' special speaking or writing styles (i.e. Trump twitter generator), publishing different versions of the same passage or information to reach diverse target audiences (i.e. versions of Bibles), and transforming a genre to make it fit on different platforms (i.e. twitter is shorter and much more casual than news).

To the best of our knowledge, existing studies on style transfer are either under supervised approach [31] or based on matching phrases extracted from corpora [19]. However, in most cases, parallel corpora between two styles are usually hard to find and it is difficult to collect enough parallel or pair-wise data for supervised training. We attempt to resolve such problems by introducing a novel sequence-to-sequence (Seq2Seq) based deep neural network model.

In this paper, we propose an unsupervised method to automatically transfer the style of a text (from early modern English employed by William Shakespeare to modern English) while at the same time retain the text's original semantic content.

The contributions of our works are as follows:

(1) We adopt an auto-encoder deep neural network model with long short-term memory (LSTM) [12] units to enable the encoder to learn a context vector and to decode the vector to words with a specific style;

(2) We introduce two switches with tensor product to control the source and the target styles, which enable unsupervised learning to be implemented in style transfer task;

(3) Unlike previous studies that failed to completely transform the text style, the style transfer method proposed in this paper transforms text's style not only on the lexical, phrasal, or syntactic levels, but also on higher levels including individuality and genre. The transformation is considered as a whole.

The rest of the paper is organized as follows. Section 2 briefly introduces the Seq2Seq deep neural network model and existing studies related to text style transfer. Section 3 describes our proposed Seq2Seq deep neural model for text style transfer (TSTSeq2Seq) that includes four important parts, i.e. switch, encoder, decoder and learning. Section 4 illustrates the details of the experiment, and analyzes the results quantitatively and qualitatively. The conclusions are given in Section 5.

2 Seq2Seq Model

The Seq2Seq model has become increasingly popular with its applications to various NLP tasks, such as machine translation [1], speech recognition [18], and dialogue systems [27] producing promising results.

A Seq2Seq model is a recurrent neural network (RNN) [21], which can take a sequence as input and generate a desired sequence. Two most widely used RNN are long short-term memory (LSTM) [12] and gated recurrent unit (GRU) [7].

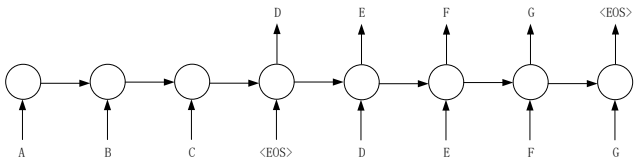

Fig. 1. A basic sequence-to-sequence model consists of two recurrent neural networks: an encoder and a decoder.

The Seq2Seq model was first proposed by Sutskever et al.[30] for machine translation. In the model as shown in the Fig. 1, two LSTMs are concatenated as an encoder and a decoder respectively, as LSTM can successfully deal with data with long range temporal dependencies. The target input information (i.e., A B C) will be encoded into an condensed context vector, then the decoder decodes the vector into a sequence of target information as output (i.e., D E F G). The whole process under the Seq2Seq model requires much less human efforts and hand-crafted feature engineering, than the state-of-the art statistical machine translation system Moses, proposed by Koehn et al. [17]. Attention mechanism was later introduced by Bahandatta et al. [1] , promoting the model's effectiveness. Seq2Seq model has achieve promising performance in many other tasks besides machine translation.

While paraphrase recognition [28, 32, 16] employing classification techniques such as RAEs, CNNs or RNNs has achieved competitive performance, generating paraphrases is left out of the picture. The generation of paraphrases can be divided into a sequence to sequence learning problem. Nevertheless, generating paraphrases requires far more efforts than operating deep neural networks applications such as image artistic style transformation. Studies such as [25, 13] were among the first to introduce the Seq2Seq model [1] directly to the task of paraphrasing. Cao et al. [6] proposed a Seq2Seq model combined with a copying decoder and a restricted generative decoder to locate the position needed to be copied and to limit the output in the source-specific vocabulary respectively. Prakash et al. [23] initially explored deep learning models for paraphrase generation, proposing multiple stacked LSTM networks by introducing a residual connection between layers. Their proposed models help retain important words in the generated paraphrases.

Similar to the task of style transfer in paraphrasing, Liu et al. [19] proposed an approach to use anchoring-based paraphrase extraction and recurrent neural networks. However, paraphrase replacement in this model partly depends on PPDB paraphrases and can only replace words in a relatively stable position without high-level style transfer on syntactical structure level.

3 The Proposed Model

The Text Style Transfer Seq2Seq model (TSTSeq2Seq) we proposed introduces two switches and some improvements into the general Seq2Seq (encoder-decoder) structure, as illustrated in Fig. 2.

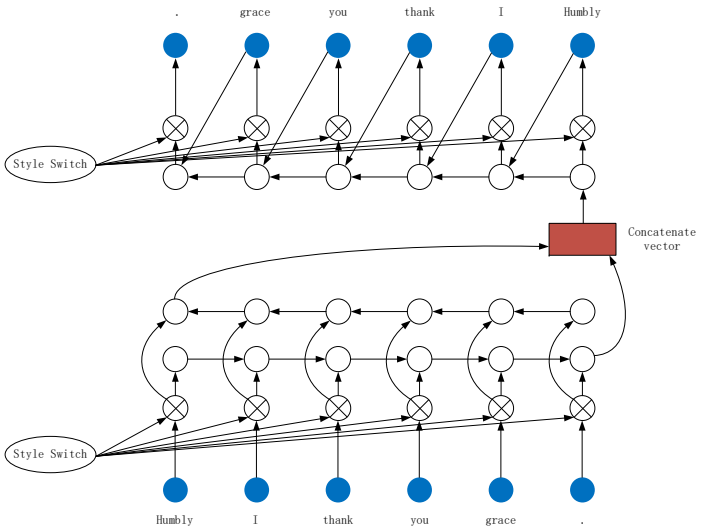

Fig. 2. The architecture of TSTSeq2Seq model

According to Fig. 2, the source text sequence first goes through a switch that embeds the style of the input text. A RNN encoder transforms the information accessed by the switch into a context vector. The context representation is then processed by a RNN decoder and by the second switch that determines the style of the output information. Finally, the target sequence is generated. The details are as follows.

We denote s as a sequence of inputs $s = \{w^1, w^2, ..., w^N\}$, where N denotes the length of the sentence. Each sentence ends with a token "$ends$" . The word w is associated with a M-dimensional embedding e_w where $e_w = \{e_w^1, e_w^2, ..., e_w^M\}$. Let V denote the vocabulary size. Each sentence s is associated with a M-dimensional representation e_s, $e_s = \{e_{w^1}, e_{w^2}, ..., e_{w^N}\}$.

An autoencoder is a neural model where the output units are identical to or directly connected with the input units. Inputs are accessed into a compressed representation by encoding, which is then used to reconstruct it back by decoding. For a sentence autoencoder, both the input X and the output Y are the same sentence s. Four important parts to the proposed model are as follows.

Switch We introduce a style switch that consists of a tensor product to enable the model to conduct unsupervised learning. The style switch is necessary in our model because pair-wise data is insufficient to conduct supervised learning and unsupervised learning requires a switch to enable the separately training of the autoencoder for two different style of text. Since we seek to transfer a text from one style (a) to another (b), the tensor is set at two values $\mathbf{s_i}$, with i being either a or b. The switch is $\mathbf{s_a} = (0, 1)$, when the input sequence is in stye a. The switch will turn into $\mathbf{s_b} = (1, 0)$, when the input sequence is in style b. We take an example to explain how tensor product works specifically, which is illustrated in Fig. 3.

$$x_i \otimes s_a = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & x_1 \\ 0 & x_2 \\ 0 & x_3 \\ 0 & x_4 \end{bmatrix}$$

$$y_i \otimes s_b = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} y_1 & 0 \\ y_2 & 0 \\ y_3 & 0 \\ y_4 & 0 \end{bmatrix}$$

Fig. 3. an example of tensor product

$\mathbf{s_i}$ decides the input information to be either on the left or the right during a tensor product function, and the weight between the switch and the input of LSTM will be trained accordingly. Therefore, the output of the switch is as follows:

$$x_t^{wi} = W_{se}(s_i \otimes e_t^{wi}), i \in [a, b] \ . \tag{1}$$

Where e_t^{wi} and x_t^{wi} denotes embedding for word and input for LSTM at position t respectively. The subscripts in Equation (1) indicate time step t, and the superscripts indicate operations of word in style wi. As a result of the operation of tensor product of style switch s_i and word in style i, weight matrices W_{se} between the switch and the input of LSTM is trained separately without overlapping along the changes in the switch.

Encoder We build the encoder based on Sutskever's work [30]. Specifically, we use LSTM as the recurrent unit, which often gains a better performance compared to the vanilla RNN. The Bi-directional Recurrent Neural Network (BRNN) [26] is introduced to ensure that the output layer is aware of the contextual information from both of the future and the past states. Then, we concatenate the

last states of both forward direction and backward directions together as concatenate vector, which conveys the compressed context information. The encoder is built as follows:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \overline{c_t} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \cdot \begin{bmatrix} \overrightarrow{h_{t-1}} \\ x_t^{wi} \end{bmatrix} . \tag{2}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \overline{c_t} . \tag{3}$$

$$\overrightarrow{h_t^{wi}}(enc) = o_t \cdot c_t . \tag{4}$$

For simplicity, we define $LSTM(x_t^{wi}, \overrightarrow{h_{t-1}}(enc))$ to be the $LSTM$ operation on vectors x_t^{wi} and $\overrightarrow{h_{t-1}}(enc)$ to achieve $\overrightarrow{h_t^{wi}}(enc)$ as in Equations (2), (3) and (4). Then we obtain:

$$\overrightarrow{h_t^{wi}}(enc) = LSTM(x_t^{wi}, \overrightarrow{h_{t-1}^{wi}}(enc)) . \tag{5}$$

$$h_s = [\overrightarrow{h_N^{wi}}(enc); \overleftarrow{h_1^{wi}}(enc)] . \tag{6}$$

In Equation (6), h_s is a concatenate vector of the last states of BRNN in two directions and is regarded as the input to the decoder.

Decoder As with encoding, the decoding operates on a unidirectional RNN with LSTMs. LSTM outputs at word level for time step t after the control of the switch are obtained by:

$$h_t^{wi}(dec) = LSTM(x_t^{wi}, h_{t-1}^{wi}) . \tag{7}$$

$$y_{t-1}^{wi} = W_{sd}(s_i \otimes h_{t-1}^{wi}(dec)), i \in [a, b] . \tag{8}$$

$$p(w|\cdot) = soft\max(e_w, y_{t-1}^{wi}) . \tag{9}$$

During decoding, the initial time step for Equation (7) is $h_0^{wi}(d) = h_s$. $LSTM$ word-decoding generates a word token $h_t^{wi}(dec)$ along every time step sequentially. The embedding is then combined with earlier hidden vectors after the switch for the next time step prediction until the ends token is predicted.

Learning The maximization likelihood estimation (MLE) is used to infer model parameters. Similar to most existing Seq2Seq models, we use cross entropy (CE) to measure the difference of probability distributions. The error of which is back propagated through LSTM. We then apply Adam optimizer [15] with mini-batches to fine tune the weights of the model, as Adam is a popular optimizer to train RNN.

During the training stage, we simultaneously set switches both in encoding and decoding layers to $S_a = [0, 1]$ when the style of the input text is a, and set both switches to $S_b = [1, 0]$ when the style of the input text is b. The switches are simultaneously assigned according to the style of the input.

4 Experiments

We use Shakespeare's plays as training and testing data for the task of paraphrasing owing a specific writing style. These plays are regarded as the most highly-regarded pieces of old English literature and which were written by a famous writer dated from 400 years ago. There are many linguistic resources available online to facilitate our research. We collected 17 Shakespeare plays from Sparknotes website, which is a corpus with corresponding relationship between early modern English employed by William Shakespeare and modern English.

The task is to transfer the style of Shakespeare's text into the style of modern English, while keeping the overall semantic content. In deep learning, a massive data set is usually required to feed into the neural network to get a well-trained model compared with the statistical machine translation (SMT) [5] method. Even though the corpus on Sparknotes is pair-wise, it is still too small to conduct supervised training for the utilized deep model. As a result, we separately train early modern English by Shakespeare and modern English by the autoencoder model with two different switches to control the learning of the styles.

The plays were sentence aligned after tokenizing and lowercasing, producing a total of 42,158 sentence pairs in the Sparknotes data. To utilize the dataset more sufficiently, we perform cross-validation. Specifically, we randomly shuffled and divided the data set into three parts with 33,726 sentences in both styles for training, 4,216 for validation, and 4,216 for testing.

Due to the insufficient data for deep LSTMs, we adopted a two-layer autoencoder and made some improvements upon it. Each LSTM layer consists of 512 hidden neurons and the dimensionality of word embeddings is set to 512.

At the testing stage, the test data in Shakespeare style was fed into the trained model. We set the encoding switch from S_a to S_b to encode and to transfer the style at the same time. The concatenate vector we got at the last time step of encoding was a style-transferred but context-unchanged unit. And then the vector was read by the decoder and the switch was changed into S_b as well, in order to get the modern English paraphrase.

As for evaluation, traditional evaluation metrics such as BLEU score are often considered useful in evaluating the quality of automatic paraphrasing. We computed the BLEU scores of the two models, and the results are shown in

Table 1. BLEU scores of different methods for Shakespearean paraphrase.

Method	Description	BLEU
TSTSeq2Seq_1Switch	Improved Seq2Seq model with only the switch on decoding level turned on during testing using shuffled training set, validation set and testing set mentioned in the experiment part.	46.26
TSTSeq2Seq	Improved Seq2Seq model with two switches both on encoding and decoding levels turned on during testing using shuffled training set, validation set and testing set mentioned in the experiment part.	47.47

Table 1. Our model TSTSeq2Seq with two switches, turned out to achieve a score of **47.47**, higher than the score achieved by the TSTSeq2Seq model with one switch in decoding turned on during testing. Even though the score we achieved is lower than the 66.28 score achieved by the supervised model with default Moses parameters, our model provides a reasonable intuition to conduct text style transfer with non-parallel data. According to the table, it suggests that the switch in the encoding phrase can help change the style of a text and still convey the semantic content of the text to the decoder.

However, we did not use parallel texts for training, and the traditional evaluation metrics such as BLEU scores are therefore meaningless and unconvincing to us [19]. Thus, we evaluated the result of text style transfer intuitively. Examples of style-transferred output sentences as outputs are listed in Table 2.

The paraphrases sampled from the test set can be sorted into different levels of style transfer. Replacements includes changing the word from "ouse" into "revive", the phrase from "come to" into "arrived in", the syntactic from "where was this?" to "where did this happen?". Changes even happen at the general style level such as from "wherefore doth lysander deny your love, so rich within his soul, and tender me, forsooth, affection, but by your setting on, by your consent?" to "and why does lysander deny that he loves you, when he loves you so deeply? ". In some circumstances, a transferred text in modern English remains the same as the text in the style of Shakespeare's old English. For example, the sentence "good words are better than bad strokes, octavius." remains unchanged. These results can partially verify the effectiveness of the introduction of the switch and style transfer mechanism with autoencoder model in capturing contextual and structural information.

5 Conclusions

In this paper, we have explored automatic text style transfer task by targeting a specific writing style based on our proposed sequence-to-sequence LSTM model.

Table 2. Examples of style-transferred paraphrases from early modern English by Shakespeare to modern English

Input	**Output**
but where was this?	but where did this happen?
and wherefore doth lysander deny your love, so rich within his soul, and tender me, forsooth, affection, but by your setting on, by your consent?	and why does lysander deny that he loves you, when he loves you so deeply?
his son was but a ward two years ago.	his son was a minor only two years ago.
sir, octavius is already come to rome.	sir, octavius has already arrived in rome.
i see him rouse himself to praise my noble act.	i see him revive himself to praise my noble act.
where is montjoy the herald?	where is montjoy, the herald?
for, you know, pyramus and thisbe meet by moonlight.	because, you know, pyramus and thisbe meet by moonlight.
there was more foolery yet, if i could remember it.	there was even more foolishness, if i could only remember it.
o you gods!	oh, you gods!
is that his answer?	is that his answer?
good words are better than bad strokes, octavius.	good words are better than bad strokes, octavius.
where is montjoy the herald?	where is montjoy, the herald?

Given that it is impractical to collect sufficient labeled data for the training of a standard deep neural network, we adopted a novel approach in using Shakespeare's plays and their translation in modern English to separately implement unsupervised learning.

In the TSTSeq2Seq model proposed by us, two switches are introduced to control the style of the encoding and the decoding phrases. The two switches allow the model to capture the input's semantic content in a style and to decode the content into another specific output style. Our study suggests that style transfer can be conducted not only at the lexical, phrasal or syntactic levels, but also on a higher level, such as author?s individualized writing style and genre preference. The proposed model in this paper can substitute the overall style of a text at the consideration of the whole sentence while preserving the semantic meaning of the original text.

While our work on autoencoder for a sentence with a specific style is only a preliminary effort toward allowing neural models to automatically deal with text style transfer, it nonetheless suggests that neural models are capable of ex-

tracting the context information from a stylized text and applying style transfer on it.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Barzilay, R.: Information fusion for multidocument summarization: paraphrasing and generation. Ph.D. thesis, Columbia University (2003)
3. Beltagy, I., Roller, S., Boleda, G., Erk, K., Mooney, R.J.: Utexas: Natural language semantics using distributional semantics and probabilistic logic. SemEval 2014 p. 796 (2014)
4. Bjerva, J., Bos, J., Van der Goot, R., Nissim, M.: The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. Proceedings of SemEval (2014)
5. Brown, P.F., Cocke, J., Pietra, S.A.D., Pietra, V.J.D., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Computational linguistics 16(2), 79–85 (1990)
6. Cao, Z., Luo, C., Li, W., Li, S.: Joint copying and restricted generation for paraphrase. arXiv preprint arXiv:1611.09235 (2016)
7. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
8. Ganitkevitch, J., Callison-Burch, C., Napoles, C., Van Durme, B.: Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1168–1179. Association for Computational Linguistics (2011)
9. Ganitkevitch, J., Van Durme, B., Callison-Burch, C.: Ppdb: The paraphrase database. In: HLT-NAACL. pp. 758–764 (2013)
10. Gatys, L.A., Ecker, A.S., Bethge, M.: Image transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2414–2423 (2016)
11. Han, L., Kashyap, A., Finin, T., Mayfield, J., Weese, J.: Umbc ebiquity-core: Semantic textual similarity systems. In: Proceedings of the Second Joint Conference on Lexical and Computational Semantics. vol. 1, pp. 44–52 (2013)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
13. Hu, B., Chen, Q., Zhu, F.: Lcsts: A large scale chinese short text summarization dataset. arXiv preprint arXiv:1506.05865 (2015)
14. Ji, Y., Eisenstein, J.: Discriminative improvements to distributional sentence similarity. In: EMNLP. pp. 891–896 (2013)
15. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
16. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Advances in neural information processing systems. pp. 3294–3302 (2015)
17. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions. pp. 177–180. Association for Computational Linguistics (2007)

18. Li, X., Wu, X.: Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. In: Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. pp. 4520–4524. IEEE (2015)
19. Liu, G., Rosello, P., Sebastian, E.: Style transfer with non-parallel corpora `http://prosello.com/papers/style-transfer-s16.pdf`
20. Madnani, N., Dorr, B.J.: Generating phrasal and sentential paraphrases: A survey of data-driven methods. Computational Linguistics 36(3), 341–387 (2010)
21. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech. vol. 2, p. 3 (2010)
22. Post, M., Ganitkevitch, J., Orland, L., Weese, J., Cao, Y., Callison-Burch, C., Irvine, A., Callison-Burch, C., Zaidan, O.F., Callison-Burch, C., et al.: Semi-markov phrase-based monolingual alignment. In: Proceedings of EMNLP. vol. 1, pp. 166–177. Association for Computational Linguistics (2013)
23. Prakash, A., Hasan, S.A., Lee, K., Datla, V., Qadir, A., Liu, J., Farri, O.: Neural paraphrase generation with stacked residual lstm networks. arXiv preprint arXiv:1610.03098 (2016)
24. Rastogi, P., Van Durme, B., Arora, R.: Multiview lsa: Representation learning via generalized cca. In: HLT-NAACL. pp. 556–566 (2015)
25. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
26. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45(11), 2673–2681 (1997)
27. Serban, I.V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., Courville, A.: Multiresolution recurrent neural networks: An application to dialogue response generation. arXiv preprint arXiv:1606.00776 (2016)
28. Socher, R., Huang, E.H., Pennington, J., Ng, A.Y., Manning, C.D.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: NIPS. vol. 24, pp. 801–809 (2011)
29. Sultan, M.A., Bethard, S., Sumner, T.: Dls@ cu: Sentence similarity from word alignment. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). pp. 241–246 (2014)
30. 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)
31. Xu, W., Ritter, A., Dolan, W.B., Grishman, R., Cherry, C.: Paraphrasing for style. In: 24th International Conference on Computational Linguistics, COLING 2012 (2012)
32. Yin, W., Schütze, H.: Convolutional neural network for paraphrase identification. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 901–911 (2015)
33. Yu, M., Dredze, M.: Improving lexical embeddings with semantic knowledge. In: ACL (2). pp. 545–550 (2014)