Seq2seq with Attention and Transformer Models of Chatbot

Anonymous ACL submission

Abstract

In this project we designed and developed a chatbot based on the Generative Models. By using the data from Cornell Movie–Dialogs Corpus, training the data in a sequence to sequence (Seq2Seq) model from TensorFlow combined with attention mechanism and Transformer, this project would be able to reply the input by generating a new comment close to the natural human-speaking way in a real-time conversation. Besides, in order to figure out how would different words embedding methods influence Seq2Seq model, and if seq2seq with attention mechanism or Transformer works better, beyond the basic model, different alternatives was added and then evaluated by BLEU. The result shows that GloVe-300d performs better than other sentence language models, while Transformer model beats the seq2seq with attention model.

1 Background

To understand and represent words of the document is the most critical foundation of any natural language processing (NLP) task, such as designing a chatbot. Researchers already realized regarding words as discrete and distinct symbols is insufficient since it ignores the semantic and syntactic similarities in the vocabulary (Levy and Goldberg, 2014). Thus, representing words based on the distributional hypothesis of Harris (1954) became a popular topic among NLP studies. Recently, methods which is known as word embedding, designed to represent words as dense vectors are proposed and believed to perform better compared with other words representations (Turian et al., 2010).

1.1 Word Embedding

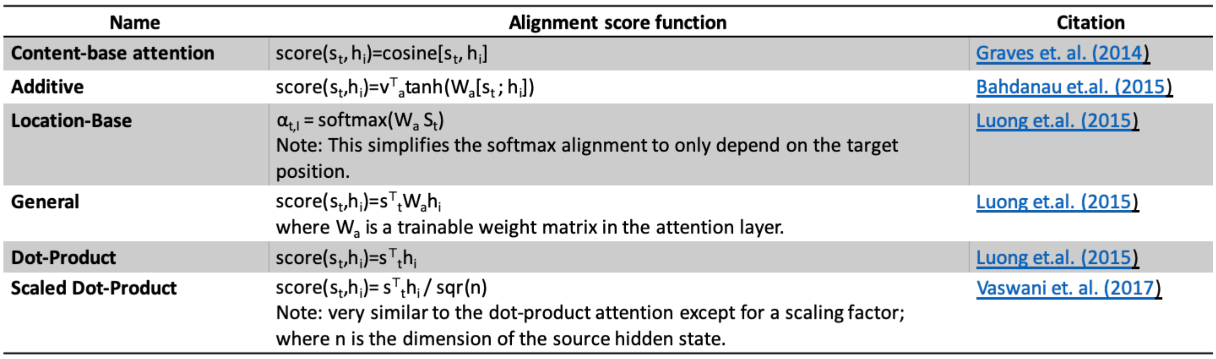
Words embedding is used to map vocabulary of the document into vectors within the real numbers. Word and phrase embedding, when used as the underlying input representation, have been shown to boost the performance in NLP tasks. Methods widely used to generate words embedding include: Word2Vec, Glove, fastText etc.

1.2 Seq2seq Model

The seq2seq model was first reported by Sutskever, et al. (2014) in the area of language modeling. The goal of this model is to transform an input sequence (source) to a new one (target) and both sequences can be of arbitrary lengths. The applications include machine translation between multiple languages in either text or audio, question-answer dialog generation, or even parsing sentences into grammar trees.

The seq2seq model generally composes of encoder and decoder. An encoder processes the input sequence and extracts the information into a context vector of a fixed length. This representation is expected to be a good summary of the meaning of the whole source sequence. The other part of seq2seq model is a decoder, which takes the context vector as an initialization to generate the output. Recurrent neural networks, such as LSTM or GRU, are units in the encoder and decoder of the model. However, the fixed-length context vector design limits its capability of remembering long sentences. The first part has frequently forgotten when it finishes processing the whole input. To resolve this issue, the attention mechanism applying in seq2seq model was developed by Bahdanau et al. (2015).

The attention mechanism was first created to help memorize long source sentence in the neural machine translation. Compared to the convention establishing a single context vector out of the encoders last hidden state, the special strategy designed by attention is to build shortcuts between the context vector and the entire source input. The



Name	Alignment score function	Citation
Content-base attention	score(s_t, h_i)=cosine[s_t, h_i]	Graves et. al. (2014)
Additive	score(s_t,h_i)=$v^T{}_a$tanh(W_a[s_t ; h_i])	Bahdanau et.al. (2015)
Location-Base	$\alpha_{t,I}$ = softmax(W_a S_t) Note: This simplifies the softmax alignment to only depend on the target position.	Luong et.al. (2015)
General	score(s_t,h_i)=$s^T{}_t W_a h_i$ where W_a is a trainable weight matrix in the attention layer.	Luong et.al. (2015)
Dot-Product	score(s_t,h_i)=$s^T{}_t h_i$	Luong et.al. (2015)
Scaled Dot-Product	score(s_t,h_i)= $s^T{}_t h_i$ / sqr(n) Note: very similar to the dot-product attention except for a scaling factor; where n is the dimension of the source hidden state.	Vaswani et. al. (2017)

Table 1: several popular attention mechanisms and corresponding alignment score functions.

weights of these shortcut connections are tailored for each output component. Since the context vector has covered the entire input sequence, the forgetting is not necessary to worry about. The context vector learns and controls the alignment between the source and target.

1.3 Attention Mechanism

By the utilization of the attention, the dependencies between source and target sequences are not restrained by the in-between distance. Given the considerable improvement by attention in machine translation, it rapidly got expanded into the image field (Xu et al. 2015) and researchers started exploring different other types of attention mechanisms (Luong, et al., 2015; Britz et al., 2017; Vaswani, et al., 2017). Table 1 summaries some popular attention mechanisms and corresponding alignment score functions and table 2 outlines broader categories of attention mechanisms.

1.3.1 Self-Attention

Self-attention is referring to different positions of a single sequence in order to compute a representation of the same sequence. It has been demonstrated to be effective in various applications, such as machine reading, abstractive summarization, or image description generation.

1.3.2 Soft and Hard Attention

Based on whether the attention has access to the whole image or only a patch, Xu et al. (2015) first proposed the difference between hard and soft attention. The alignment weights of soft attention are learned and implemented softly over all patches in the image, so the model is smooth and differentiable. The disadvantage is obvious that when the input is large this process is expensive. The essential of the soft attention is the same as the type of attention in Bahdanau et al., 2015.

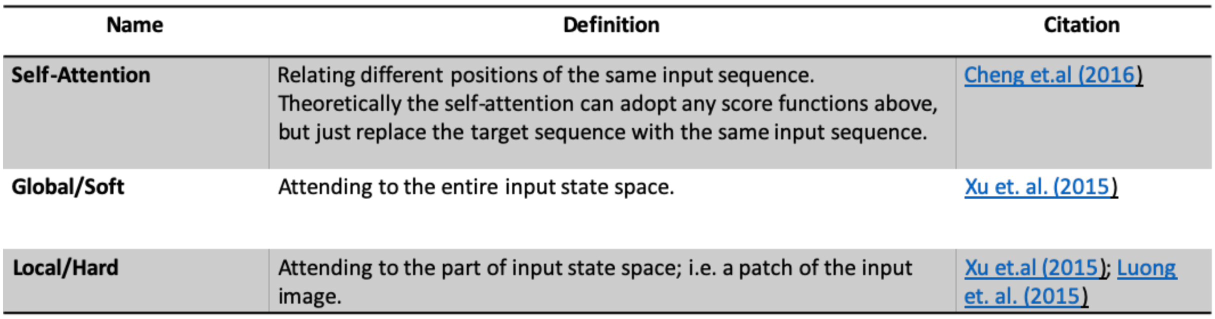
On the other hand, hard attention only pay attention on one patch of the image at a time. The merit is that it requires less calculation at the inference time, while the disadvantage is that the model is non-differentiable and requires more complicated techniques. (Luong, et al., 2015)

1.3.3 Global and Local Attention

The global and local attention were first proposed by Luong, et al. (2015). The global attention is similar to the soft one, whereas the local one is a blend of hard and soft attention. The hard attention was improved to be more differentiable: a single aligned position was predicted by the model for the current target word and the model uses a window centered around the source position to compute a context vector.

1.4 Transformer

Complex recurrent and convolutional neural networks, containing an encoder and a decoder, based sequence transduction models have been developed as a state-of-art method by numerous efforts. As mentioned by Ashish Vaswani, et. al. (2017), recurrent models have been facing critics for the longer sequence lengths, because of the constraints of memory limit batching across samples. Factorization tricks and conditional computation have been established to improve the performance in computational efficiency; however, the constraint of sequential computation is still unresolved. To address this problem, Vaswani et al. (2017) introduce self-attention mechanisms into the sequence transduction models, and the authors called this model the Transformer, which allows them to ignore the dependencies between distant positions and reduces to a constant number of operations. Self-attention executed a constant number of sequential operations, while RNN has to



Name	Definition	Citation
Self-Attention	Relating different positions of the same input sequence. Theoretically the self-attention can adopt any score functions above, but just replace the target sequence with the same input sequence.	Cheng et.al (2016)
Global/Soft	Attending to the entire input state space.	Xu et. al. (2015)
Local/Hard	Attending to the part of input state space; i.e. a patch of the input image.	Xu et.al (2015); Luong et. al. (2015)

Table 2: broader categories of attention mechanisms.

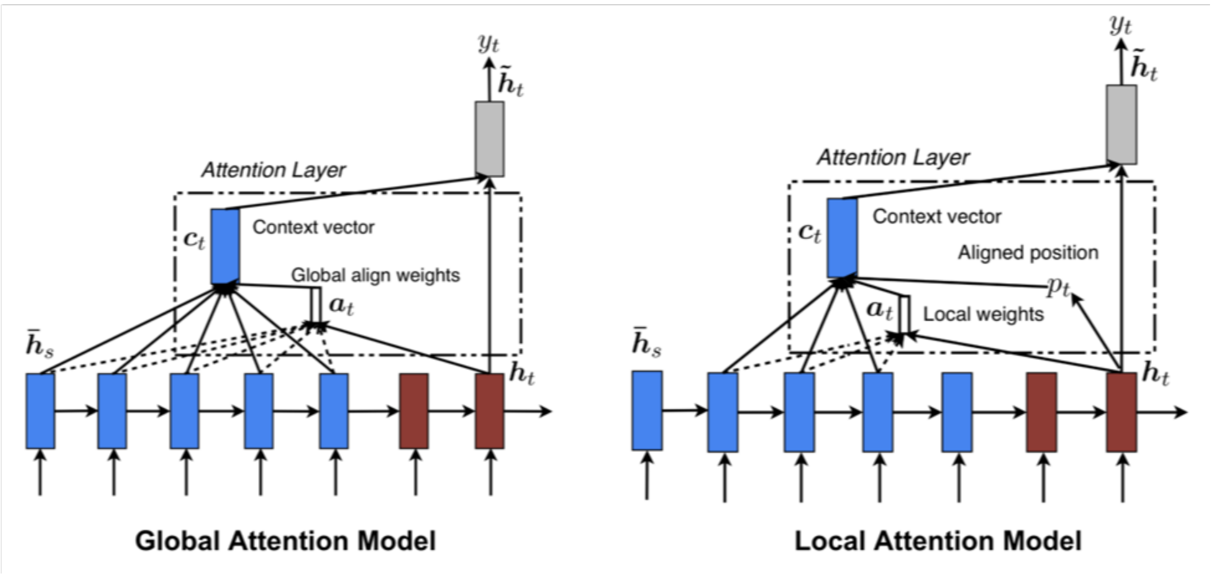


Figure 1: Global vs local attention (Image source: Fig 2 3 in Luong, et al., 2015

fulfill O(n) time operations. What is more, since machine translations always confront the circumstance that sequence length is shorter than the representation dimensionality, it is quite appropriate for the self-attention to implement in such process to improve the computational performance than RNN. Besides, models established with self-attention are more interpretable in terms of the syntactic and semantic structure of the sentences. It is the first model that compute representations of its input and output without implementing sequential RNNS or convolution layers, instead, relying on self-attention totally. The Transformer established a state-of-the-art approach in the sequence modeling and transduction issues both on accuracy performance and on computational efficiency.

2 Approach

2.1 Word Embedding

In this study, word2vec and Glove with different dimensions were applied to generate words embedding.

Word2vec is a group of related models, it becomes quite popular since it was first come up with by a team of researchers led by Tomas Mikolov in 2013. Word2vec is designed to be a two-layer neural network, taking text corpus as input and output a series of vectors. Typically, word2vec usually applies either of two models to generate a distributed words representation: continuous bag-of-words (CBOW) or continuous skip-gram. Specifically, unlike ordinary bag-of-words model, CBOW utilize continuous distributed representation of the context(Mikolov et al.,2013). In CBOW future words are used by building a log-linear classifier with four future and four history words at the input, where the training criterion is to correctly classify the current (middle) word.

The training complexity of this architecture is proportional to

$$Q = N * D + D * log_2(V)$$

In the skip-gram model, each current word is used as an input to a log-linear classifier with continuous projection layer, then the model predicts words within a certain range before and after the current word.

The training complexity of this architecture is proportional to

$$Q = C * (D = D * log_2(V))$$

where C is the maximum distance of the words.

Overall, CBOW is believed to run faster while skip-gram model performs better.

Global Vectors for word representation (GloVe) was first proposed by Pennington et al., researchers indicated that that prior studies on word embedding are poor in either doing word analogy task or using the statistics of the corpus (2014). In this case, a weighted least squares model that trains on global word-word co-occurrence counts and thus makes efficient use of statistics was conducted. To build up the model, first is to establish a matrix of word-word co-occurrence counts, then to remove those words rarely occur, besides, a weighted least squares regression function is added. The final model is shown as below:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \widetilde{w}_j + b_i + \widetilde{b}_j - logX_{ij})^2$$

Where V is the size of vocabulary, $f(X_{ij})$ is the weighting function.

According to the authors, this model is convinced to perform better on words analogy task than prior models. What is more important, to compare with word2vec, GloVe outperforms word2vec within the same corpus, vocabulary, window size, and training time.

2.2 Global Attention

The context vector consumes three pieces of information. There are encoder and decoder hidden states and alignment between source and target. The equations of this model is described as in Table 1. The architecture of Seq2Seq model (Figure 2) with attention is shown in the Eigure 3.

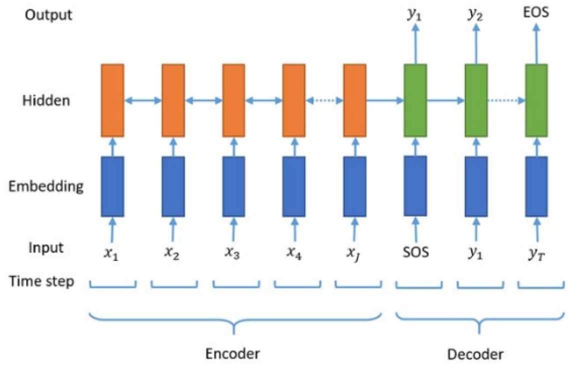


Figure 2: The basic seq2seq model. SOS and EOS represent the start and end of a sequence, respectively.

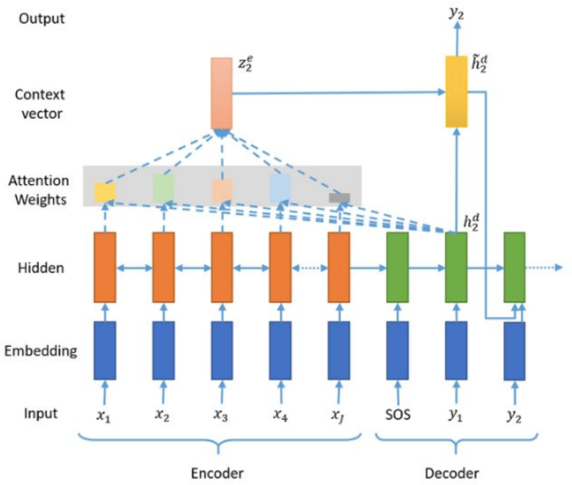


Figure 3: An attention-based seq2seq model.

2.3 Transformer

The proposed model is entirely built on the self-attention mechanisms without using sequence-aligned recurrent architecture. The major component in the transformer is the unit of multi-head self-attention mechanism.

the encoded representation of the input as a set of key-value pairs, (K,V), both the keys and values are the encoder hidden states. In the decoder, the previous output is compressed into a query (Q of dimension m) and the next output is produced by mapping this query and the set of keys and values.

The transformer adopts the scaled dot-product attention: the output is a weighted sum of the values, where the weight assigned to each value is determined by the dot-product of the query with all the keys:

Attention(Q, K,V)=softmax(QKT/sqr (n))V

The full Architecture was shown as Figure 4.

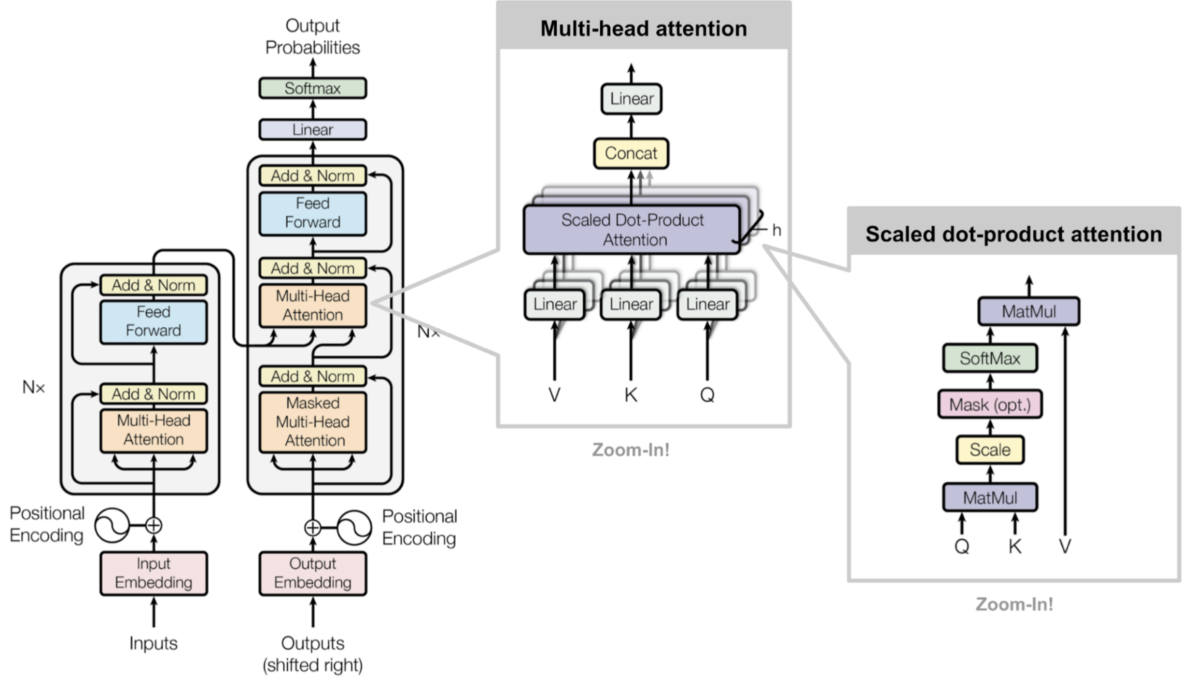


Figure 4: The full model architecture of the transformer. (Image source: Fig 1 2 in Vaswani, et al., 2017.)

2.4 Evaluation

To evaluate how well the chatbot model works, we consider using the word-overlap evaluation to examine the quality of proposed response compared to the ground-truth response. There is several word-overlap evaluation metrics to be generally used, and we took BLEU in this project. BLEU analyses the overlap rate of n-grams in the ground truth and the proposed responses.

Because of different length of n-grams is considered, BLEU is also shown as BLEU-N, where N means the maximum length. In this project, we defined N as 1,2,3,4, respectively, which means 1,2,3, and finally 4 grams is used to help the evaluation.

Although as opposed to machine translation, which commonly uses the cumulative scores of BLEU-1, BLEU-2, BLEU-3, and BLEU-4 to evaluate the generated sentence, BLEU-2 has been noted to be closely associated with human annotators among the BLEU metrics and most frequently used in chatbot evaluation(Papineni et al., 2002), in this project we still evaluated the model with n-gram changes from 1 to 4 in order to obtain a more detailed result.

3 Experiments

3.1 Data Set

For this project, we are using Cornell Movie–Dialogs Corpus as the data. This dataset contains 220,579 conversational exchanges between 10,292 pairs of movie characters from 617 which is in total 304,713 utterances.

This dataset is posted online as a public resource, frequently used among NLP tasks.

3.2 Pre-trained Words Embedding

In order to save time and obtain a better performance, in this project we used pre-trained words embeddings.

For word2Vec section, the pre-trained vectors we used is published by Google, which is trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases.

For GloVe section, we used pre-trained data training from Wikipedia 2014 and Gigaword 5 (about 400,000 vocabs in total), which contains 50d, 100d, 200d, and 300d vectors.

3.3 Build Seq2Seq model with word2vec

To build the model, we first cleaned movie–dialogs corpus data by sorting the question-answer pairs, removing unnecessary characters and altering the format of words. In order to predict in a more appericate way, we then filtered out corpus too small or too long.

Restricted by the limited RAM capacity, in this section we used only 1500 question-answer pairs as the data, where 80% of it was used for training, the rest was used for validation.

After selecting the data, all the sentences were toked and stored in the vocabulary, and words rarely occur were removed. Besides, dictionaries for both encoding and decoding that would save the vector for each word were created.

With all preparation, the pre-trained word2vec data was then loaded. By using KeyedVectors from gensim.models we read and stored the words and vectors from the bin. file, and build the encoding and decoding matrices.

Now, the word embedding is ready to feed our neural network with attention mechanism. Details of how the neural network was build would be discussed in other sections since here we focus on the performance of different words embedding methods.

3.4 Build Seq2Seq model with GloVe

Similar as the prior word2vec section, we first did prepossessing for the data. In order to control the variable and compare the result with the same conditions, we randomly took 1500 pairs of data and separate into training and validation sets.

To differ with loading pre-trained word2vec embedding, in this section we simply store the words and vectors respectively into dictionaries, then filled the encoding matrix and decoding matrix.

Besides, in order to reveal effect of different dimensions within the same words embedding, in this section, pre-trained GloVe words embedding varies from 50 dimentions to 300 dimensions would be loaded and evaluated.

As these two sections were designed to compare the performance of different words embedding methods, here we used the same neural network model as the word2vec used before.

As shown in Table 1, generally BLEU-1 get the highest scores for all different words embedding methods with different dimensions among the BLEU metrics. Although BLEU-2 is believed to perform better earlier due to it is close to natural language speaking habits. In this project BLEU-1 would be a better method to access the chatbot.

Generally, Glove words embedding with higher dimensions performs better than lower one. For 50 dimensions, model only gained 2.73 in BLEU-1, while with 300 dimensions GloVe words embedding, BLEU-1 scores increased to 3.34.

Words Embedding	BLEU-1	BLEU-2
GloVe-50d	2.73	2.28
GloVe-100d	2.90	2.50
GloVe-200d	3.10	2.60
GloVe-300d	3.34	3.27
word2vec-300d	2.26	1.62
Words Embedding	**BLEU-3**	**BLEU-4**
GloVe-50d	0.90	2.73
GloVe-100d	0.94	2.90
GloVe-200d	1.00	3.10
GloVe-300d	1.12	3.34
word2vec-300d	0.67	2.26

Table 1: BLEU-N Results

Besides, compared to word2vec, GloVe with the same dimensions performs much better in this project, which gained 1.12 in BLEU-3 scores almost as twice as word2vec's. In addition, even GloVe words embedding with lower dimensions, such as 50 dimensions, also overperformed word2vec with 300 dimensions.

However, for both word2vec and Glove with different dimensions, all the words embedding got BLEU-4 scores as same as BLEU-1, which could probably due to the error in calculating BLEU-4.

3.5 Build Seq2Seq model with attention

The model was established with global attention. Based on the seq2seq model, we add a global layer on the model, train the model with the movie dialogue pairs, batch size is 64 for 50 epochs.

3.6 Build Transformer model

Both encoder and decoder contain a stack of N = 4 identical layers, Each layer has a multi-head self-attention layer and a simple position-wise fully connected feed-forward network. Both the source and target sequences first go through embedding layers to produce data of the same dimension dmodel=512. A softmax and linear layer are added to the final decoder output.

4 Conclusion

In conclusion, Glove performed better than word2vec in chatbot model, even with lower dimensions. Although word2vec is easier to understand and apply for freshmen, the sub-linear

Chatbot Models	BLEU-1	BLEU-2
Seq2Seq	1.32	1.21
Seq2Seq + Attention	2.97	2.88
Transformer	3.42	2.96

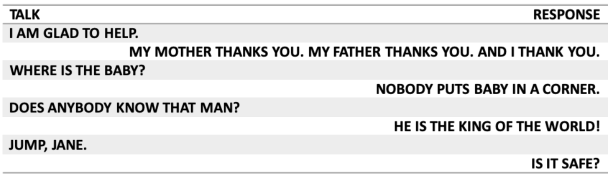
Table 2: BLEU-1,2 Results

relationships are not explicitly defined compared with Glove. Besides, as GloVe considers relationships between word pair and word pair rather than word and word, at the meanwhile, wights less for highly frequent word pairs so as to prevent the meaningless stop words like the, an will not dominate the training progress, GloVe is able to generates answers closer to the ground truth data than word2vec.

The dimension of words embedding actually refers to the length of these vectors. When words are embed into space, those words have similar meaning are usually close to each other, that is, have similar vector representations, in this case, words embedding with higher dimensions is more possible to differ them since they are able to contain more information.

Compared with the simple seq2seq baseline model, both seq2seq with global attention and transformer models has significantly improved the performance. Although Transformer model only uses 4 layers on both encoder and decoder parts, it still beats other models. Transformer has a vital implement in pre-trained models such as BERT and GPT-2. The powerful Transformer will have more wide applications in images, voice and videos et. al.

5 References



TALK	RESPONSE
I AM GLAD TO HELP.	
	MY MOTHER THANKS YOU. MY FATHER THANKS YOU. AND I THANK YOU.
WHERE IS THE BABY?	
	NOBODY PUTS BABY IN A CORNER.
DOES ANYBODY KNOW THAT MAN?	
	HE IS THE KING OF THE WORLD!
JUMP, JANE.	
	IS IT SAFE?

Figure 5: Samples of chat dialogues from Transformer model

[1] Attention and Memory in Deep Learning and NLP. - Jan 3, 2016 by Denny Britz.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. ICLR 2015.

[3] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. ICML, 2015.

[4] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. NIPS 2014.

[5] Thang Luong, Hieu Pham, Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. EMNLP 2015.

[6] Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. ACL 2017.

[7] Ashish Vaswani, et al. Attention is all you need. NIPS 2017.

[8] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. EMNLP 2016.

[9] Xiaolong Wang, et al. Non-local Neural Networks. CVPR 2018.

[10] Tomas Mikolov, et al. Distributed Representations of Words and Phrases and their Compositionality. Neural information processing systems, 2013.

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014

[12] Omer Levy and Yoav Goldberg. Dependency-Based Word Embeddings. Proceedings of the 52nd Annual Meeting of the Association for Computational, 2014.

[13] Zellig S. Harris. Distributional Structure. WORD, 1954.

[14] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word Representations: A simple and general method for semi-supervised learning. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010.