2020 9th International Congress on Advanced Applied Informatics (IIAI-AAI) | 978-1-7281-7397-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/IIAI-AAI50415.2020.00029

Text Entailment Generation with Attention-based Sequence-to-sequence Model

Xiaomei Zhao
Graduate School of Humanities and Sustainable Science
Osaka Prefecture University
Osaka, Japan
mdb04019@edu.osakafu-u.ac.jp

Hidekazu Yanagimoto
Graduate School of Humanities and Sustainable Science
Osaka Prefecture University
Osaka, Japan
hidekazu@kis.osakafu-u.ac.jp

***Abstract*—Text entailment needs semantic similarity judgment between two sentences and is a good task to measure text understanding. If we realize entailment generation, we can apply it to summarization that keeps semantics between an original text and a generated text. These days, neural networks are employed to construct modules that encode an original text and generate summarization. In natural language processing, sequence-to-sequence models, which realize sequential learning, are employed to develop machine translation. Moreover, attention mechanism is proposed to improve machine translation considering word alignment between a source language and a target language. In this paper, we applied an attention-based sequence-to-sequence model to an entailment generation task and confirmed the system realized entailment generation. The proposed method can capture important words in the input text and generate a frequent sentence, which is grammatically correct and semantically appropriate. The results mean the proposed system understands a text semantically.**

Keywords—Entailment Generation, Attention mechanism, Sequence-to-sequence model

I. INTRODUCTION

Text entailment[1] is an important task in natural language inference because it needs semantic similarity judgment between two sentences. Moreover, it can be applied to many applications such as chatbot, question-answering systems, and so on. In entailment tasks, the system has to judge whether an inference sentence is consistent with a premise sentence or not. For example, we discuss the below sentences.

(1) He went to the library again to borrow books.

(2) This is not his first visit to the library.

We can inference Sentence (2) from the premise Sentence (1). In this case, the relation between Sentence (1) and Sentence (2) is an entailment linguistically. The judgment needs the ability of understanding sentences. We propose an entailment generation task that generates an inference sentence from a premise sentence.

To realize entailment generation, an entailment generation system has to understand a premise sentence and generate an inference sentence based on the semantic of the premise sentence. In entailment tasks, only an inference sentence is prepared for a premise sentence. On the other hand, in entailment generation tasks, some inference sentences can be generated for a premise sentence and all the inference sentences are acceptable. For example, the next inference sentence is suitable for Sentence (1).

(3) He went to borrow books.

So, we have to consider multiple inference sentences from only a premise sentence.

The process for entailment generation is similar to the process for machine translation because both of the systems need to understand an input sentence and generate an output sentence based on the input sentence. A sequence-to-sequence model is often used to construct a machine translation system[2] Moreover, an attention mechanism improves machine translation precision and is combined with a sequence-to-sequence model[3][4]. Entailment generation can be applied to various tasks such as chatbot, question-answering systems.

In this paper, we proposed an entailment generation system with an attention-based sequence-to-sequence model. The model achieves the highest performance in machine translation but nobody knows whether the model can achieve the highest performance in entailment generation. So a goal is to confirm whether the proposed method can generate an appropriate inference sentence from a premise sentence. Another goal is to generate multiple inference sentences from a premise sentence. To generate multiple inference sentences from one premise sentence, we change the first word of the inference sentence in generating it. When the first word is generated in the output, we keep two words with the highest softmax score. Then, the system receives the two different words as the initial words and generates two output sentences. In the end, we got two inference sentences.

In experiments, we use the corpus of Stanford Natural Language Inference (SNLI). We train the proposed system and generate two inference sentences from a premise sentence in SNLI. From the experiments, we found that the proposed method can generate inference sentences from input sentences considering entailment. Moreover, we generate another inference sentence from the same premise sentence with the initial word selection strategy. In this case, we found that the system could generate a sentence that describes a different aspect of the input sentence.

978-1-7281-7397-9/20/$31.00 ©2020 IEEE
DOI 10.1109/IIAI-AAI50415.2020.00029

In this paper is constructed below. In Section II, we describe related works. Especially, we explain a sequence-to-sequence model and an attention mechanism. In Section III, we explain the proposed system that generates an inference sentence considering entailment. In Section IV, evaluation experiments execute with the corpus of Stanford Natural Language Inference. We discuss the proposed method by comparing an inference sentence with a premise sentence. In Section V, we describe conclusions and future works.

II. Related Works

A. Machine translation

Machine translation[5] is the main topic of natural language processing and one of text generation tasks. In a traditional approach, statistical machine learning is employed to translate a source language to a target language. Especially, sequential learning is often employed because translation is regarded as a transformation from a sequential data to another sequential data.

Many researchers pay attention to deep learning and apply it to machine translation. In machine learning, sequence-to-sequence models are employed. The sequence-to-sequence model consists of an encoder, which makes a real vector from an input sentence, and a decoder, which makes an output text from the real vector. However, only a real vector is sent from the encoder to the decoder in this model and it is impossible to send enough information from the encoder to the decoder. To remedy the problem, an attention mechanism is proposed and improves machine translation accuracy.

B. Sequence-to-sequence model

A sequence-to-sequence model was proposed by Sutskever et al.[2]. The model consists of an encoder and a decoder and both of them is constructed with a recurrent neural network. In natural language processing, a language model predicts the next word based on preceding words and the language model is called an n-gram model[5]. The recurrent neural network can emulate the n-gram model and capture the structure of natural languages. Moreover, recurrent neural networks can accept a variable length text and generate a variable length text.

Figure 1 shows the model of sequence-to-sequence proposed by Sutskever et al.[2]. The left of the dotted line is the encoder, and the right of the dotted line is the decoder. The encoder reads an input of "ABC" , and then the decoder generates an output sentence "abcd". In the encoder, the model reads the input words in chronological order. By Long Short-term Memory(LSTM)or Gate Recurrent Unit(GRU), the information of the whole input sentence will be transmitted to the last LSTM(or GRU) cell. And this information will be passed to the decoder. In the decoder, the model predicts the next word based on both the last word and all the information of input sentence which is from the encoder. The "<end>" is the stop token. When the encoder generates "<end>", the model stops prodecing the next word. So, the length of the output sentence can differ from the length of the input sentece.

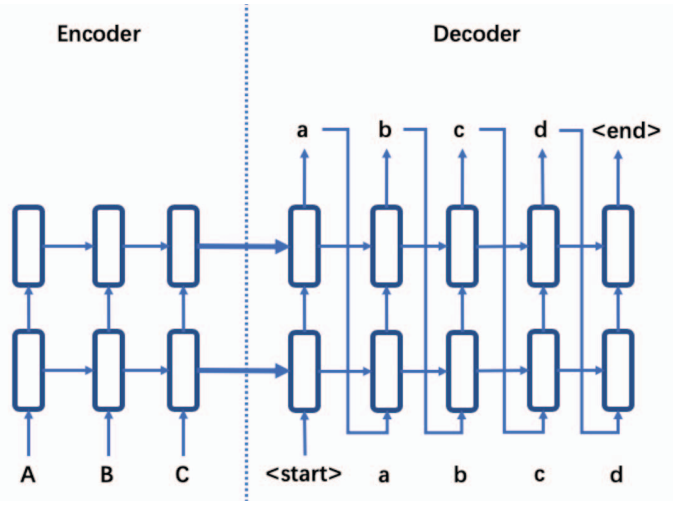

Fig.1. Sequence-to-sequence model from Sutskever et al.[2]

Simple sequence-to-sequence models cannot generate a long sentence because the vector does not have enough information, which is sent from the encoder and the decoder. One solution is attention mechanism to remedy the problem.

C. Attention mechanism

Attention mechaism is proposed by Bahdanau et al.[4] to introduce word alignment in neural machine translation. In machine translation, the word alignment means a dictionary that matches source language words and target language words. If a machine translation system can define correct word alignent, the system can translate source words into target words from the viewpoint of semantics.

Figure 2 shows the architecture of an attention-based sequence-to-sequence model propose by Loung et al.[3]. The system includes a sequence-to-sequence mode and attention mechanism. The attention mechanism generates features from all states in the encoder and the decoder generates the next word considering the states and a previous decoder state. The decoder can consider all states in the encoder regardless of the length of the output.

They add an attention layer between the encoder and the decoder. The attention layer can assign attention weights to the Long Short-term Memory(or Gate Recurrent Unit) cells of the encoder.

D. Textual Entailment Generation

Textual entailment generation is a new task to infer appropriate sentences from a given premise. Many researchers apply neural networks to the textual entailment generation.

Kolesnyk et al.[7] used an attention-based sequence-to-sequence model to realize textual entailment generation. Especially, they generated a sentence from a give premise with greedy decoding. Guo et al.[8] proposed an attention-based sequence-to-sequence model with residual LSTM. They improved an encoder in the sequence-to-sequence model to generate a content vector including more premise information.

Moreover, Pasunuru et al.[9] applied entailment genration to caption generation. Especially, they used attention mechanism to realize entailment generation for caption generation.

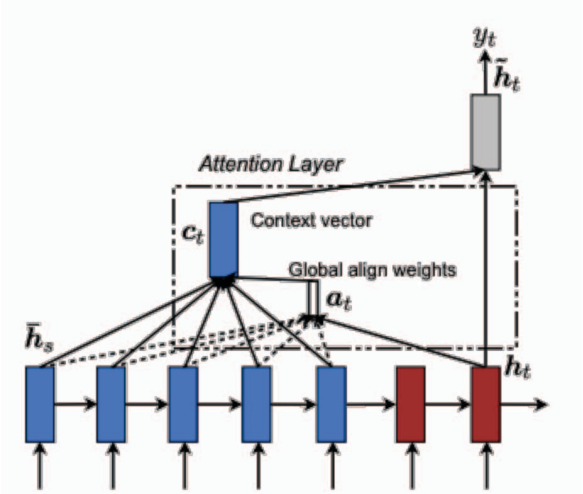

Fig. 2. Global attention based sequence-to-sequence model from M. Loung et al [1]

III. Entailment Generation

We proposed an entailment generation system, which is implemented with a sequence-to-sequence model with global attention. The global attention means attention weights are calculated considering all states in the encoder.

To generate multiple outputs from one sentence considering entailment relationship, we change the first word in the decoder at the prediction stage. When the first word is generated, we pick up the two words with the highest softmax score. Using the two words as the initial words, the system generates two sentences based on one input sentence.

A. Entailment generation

In entailment generation, the system receives a premise sentence as an input and generates an inference sentence as an output. The task is regarded as transformation from an input sequential data to an output sequential data. In this paper, we employ a machine translation system to develop an entailment generation system. Speaking concretely, we use a sequence-to-sequence model.

The encoder in a sequence-to-sequence model receives an input sentence and transforms words in the sentence into real vectors. After then, a recurrent neural network embeds each word into real vectors considering surrounding context. In this paper, we used Gate Recurrent Unit as a recurrent neural network.

The decoder in a sequence-to-sequence model set the final hidden state in the encoder as the initial hidden state. The hidden layer in the decoder is implemented with Gate Recurrent Unit. The initial word for the decoder is special symbol that denotes beginning of a sentence. After then, inputs are words predicted by the decoder.

The attention layer is between the encoder and the decoder. The function of the attention layer is to calculate attention weight, which means importance of states of the encoder in prediction, and to generate an attention-based context vector, which denotes information to predict the next word in the decoder. To get attention weight (a_t), we calculate the score between h_t *and* $\bar{h}_s$. And h_t is the hidden state of the decoder at the current time, $\bar{h}_s$ is a set of the hidden state of the encoder at all time. The score is calculated with a fully connected neural network. After the score is obtained, softmax operation is performed on the score to obtain attention weights. The context vector is calculated as the weight sum of hidden states in the encoder based on the attention weights. The context vector is the final output of the attention layer. The context vector is submitted to the decoder and the decoder predict the next words for entailment generation. This process is defined as equations below.

$$\text{score}(h_t, \bar{h}_s) = v_a^{\mathrm{T}} \tanh(W_1 h_t + W_2 \bar{h}_s) \quad (1)$$

$$a_t = \text{align}(h_t, \bar{h}_s) = \frac{\exp(score(h_t, \bar{h}_s))}{\sum_{s'=1}^{S} \exp(score(h_t, \bar{h}_{s'}))} \quad (2)$$

$$c_t = \sum_s a_t \bar{h}_s \quad (3)$$

$$\tilde{h}_t = \tanh(W_c[c_t; h_t]) \quad (4)$$

B. Multiple inference sentence generation

Generally, we can make multiple inference sentences from the same premise sentence based on entailment. So, it is favorable for the entailment generation system to make multiple inference sentences from the same premise sentence. However, when the input data is identical, machine learning cannot generate a different output because machine learning constructs a mapping function from an input to an output. To generate some different outputs, we change the initial word in the decoder.

To generate two outputs from one sentence, we change the initial word in the decoder at the prediction stage. The initial input for the encoder is a special symbol that means the beginning of a sentence. The recurrent neural network outputs a vector, $\tilde{h}_t$, and the system predict the next word based on the vector. In fully connected layer, y_t is the final output of the decoder and means an occurrence probability for vocabulary. So, it means probabilities of all words in vocabulary is calculated and we understand which word is important according to the probabilities. The probabilities are calculated with a fully connected neural network.

$$p(y_t | y_{<t}, x) = \text{softmax}(W_s \tilde{h}_t) \quad (5)$$

We can pick up some words according to the probabilities. In this paper, we select only the first prediction and obtain different outputs.

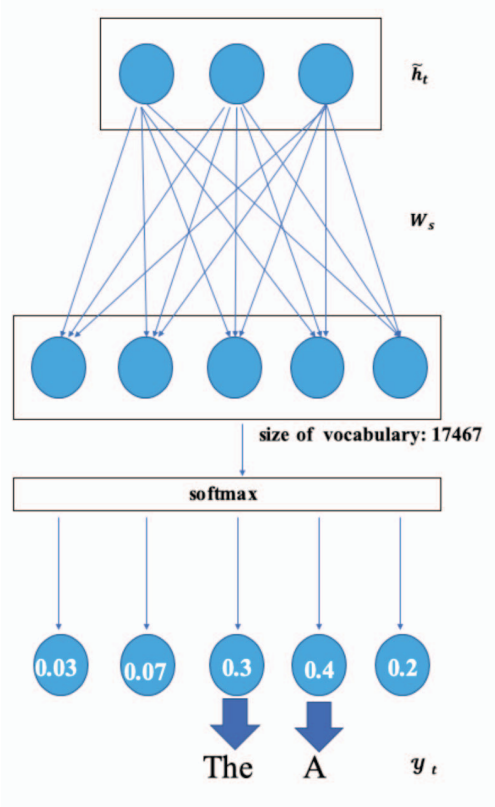

Fig. 3. How to select the initial word for multiple output generation

In Fig.3, we show the fully connected layer to generate different outputs. The $\tilde{h}_t$ is the output of the hidden state in the decoder at the current time. By multiplying $\tilde{h}_t$ and W_s, we get the probability of each word in the vocabulary. The vocabulary consists of words in training dataset and special symbols. After calculating the probabilities, we pick the two words "A" and "The" which have the highest probabilities because we think that the words are related to the input sentence strongly. In this case, the word "A" and the word "The" are selected as the initial word candidates to generate the outputs.

In Fig.4, we used the words "A" and "The" as the initial words to generate different sentences considering entailment. The input sentence is "A woman, in red and white, and wearing glasses, sits in a room with other people.", And we get two outputs:

output1: A woman is sitting down.
output2: The woman is wearing red and white.

The following words are generated with the decoder automatically. Generally, the two sentences has different semantics because the initial words are different and the hidden states in the decoder changes differently.

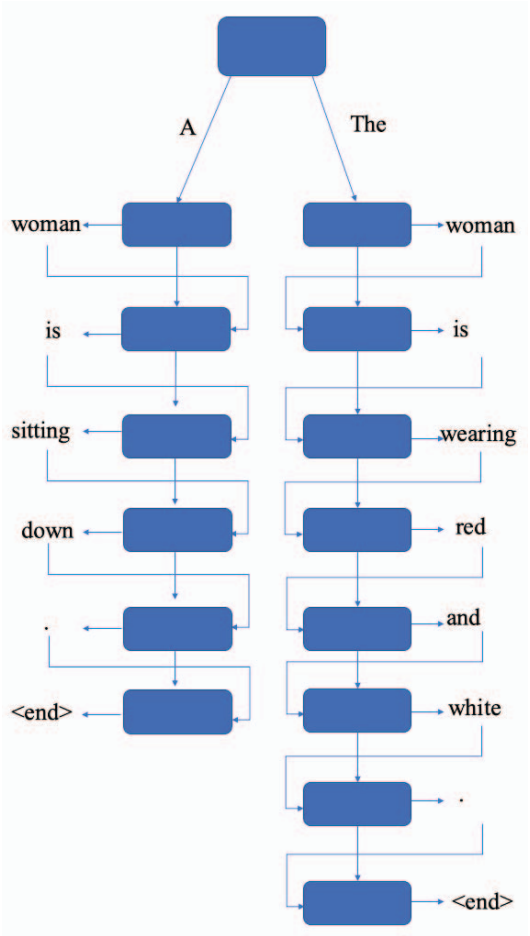

Fig. 4. generate two output by two intimal words

IV. EXPERIMENTS

In this section, we discuss the proposed method based on some experiment results. The proposed system is trained with a corpus and generates two different sentences considering entailment. After then, we confirm the proposed method is effective for entailment generation.

A. Dataset

The Stanford Natural Language Inference(SNLI) corpus is a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the label "entailment", "contradiction", and "neutral". The corpus is employed to evaluate the natural language processing tasks. In this experiment, to discuss entailment generation, we extracted pairs of sentences representing entailment meanings from SNLI as the dataset. There are 183,416 data for training and 3,368 data for testing. The size of the source vocabulary is 22,852, the size of the target vocabulary is 17,467.

B. Settings

Both of the encoder and the decoder are implemented with Gate Recurrent Unit. The Attention mechanism is global attention. the number of training loops is 30. The loss function is sparse categorical corssentropy. The optimizer is Adam. Hyperparameter settings are in the Table 1.

TABLE I.
THE PROPOSED METHOD SETTINGS

Items	Set
word embedding dim	256
batch size	64
encoder/decoder cell	Gate Recurrent Unit
GRU hidden state size	1024
attention mechanism	global attention
number of training loops	30
loss function	Sparse Categorical Crossentropy
optimizer	Adam

We used the code provided by TensorFlow[1] to generate text entailment, and added the Multiple Inference Generation function on this basis.

C. Result

In Figure 5, we show the training loss with respect to the learning epoch. After 30 epochs, the loss goes down from 0.5737 to 0.0735. This graph means the training executes correctly.

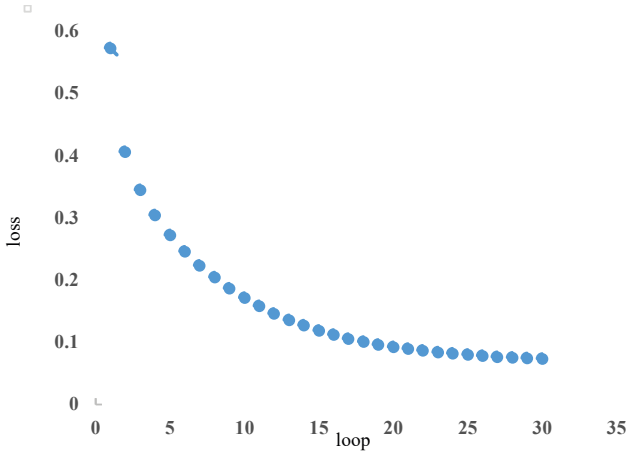

Fig. 5. loss of train

In Table II, we show the BLEU scores of two generated entailments. The BLEU scores are 36.61 and 35.59. It means the training executes correctly. But the model accuracy still needs to be improved.

TABLE II.
QUANTITATIVE EVALUATION

Items	BLEU
Output1	36.61
Ouput2	35.59

We show some predictions to verify the training results.

1) Entailment generation: In the next example, the system successfully generats the output:

(Input): A couple walk hand in hand down a street .
(Output): Two people are walking down the street .

Especially, the proposed system understands “A couple” semantically and generates “two people” in the output. Moreover, the output sentence is correct syntactically.

The next example is wrong generation.

(Input): A young family enjoys feeling ocean waves lap at their feet .
(Output): A happy couple enjoys the beach .

In the sentence pairs, the system confused the relationship between “a family” and “a couple”. Sometimes, "a family" and "a couple" has a similar meaning but it's not exactly equal. This shows that the model does not understand the premise sentence very well.

2) Two entailment generation: By changing the first word, we can get two outputs from one premise sentence. Some of them described different meanings:

(Input): A woman , in red and white , and wearing glasses , sits in a room with other people .
(Output)1: A woman is sitting down .
(Output)2: The woman is wearing red and white .

In the sentence pairs above, the system chose “A” and “The” as the initial word of each sentence. And generated entailment sentences describe different meanings from different aspect. It means that the propose method can generate a sentence based on different aspects in the premise sentence.

The next example shows the system generates the similar sentences regardless of the initial word:

(Input): Several women are playing volleyball .
(Output)1: The women are playing a game .
(Output)2: There are women playing a game .

In the sentence pairs above, the model chose “The” and “There” as the initial word of each sentence. Although the two sentences are expressed differently, they have the same meaning. To avoid this problem, we have to introduce more diversities in the propose method.

V. CONCLUSIONS

In this paper, we used an attention-based sequence-to-sequence model to generate an inference sentence from a premise sentence considering entailment. To generate multiple

[1] https://www.tensorflow.org/tutorials/text/nmt_with_attention

sentences from one input sentence, we changed the initial word in the decoder at the prediction stage. At a result, the system can generate different sentences from the same input sentence. It means that an attention-based sequence-to-sequence model realizes entailment generation well. However, some generated sentences are wrong. This shows that the model does not understand the premise sentences very well. By changing the initial word, we got two outputs from one input of the premise sentence. Some of them described different meanings, but some of them have a similar meaning. This shows that it is not enough just to change the initial word.

In the future, we will focus on improving the language understanding of the system. Especially, we improve the encoder to extract more information from an input sentence. Meanwhile, we will apply a beam search algorithm to the whole sentences to get more than two entailments, and cluster them to get multiple inference sentences from different clusters and improve diversity in entailment generation.

REFERENCES

[1] D. Z. Korman, E. Mack, "Defining Textual Entailment," Journal of the Association for Information Science and Technology, Vol. 69, No. 6, pp.763-772, 2018.

[2] I. Sutskever, O. Vinyals, QV Le , "Sequence to Sequence Learning with Neural Networks," NIPS, 2014, pp.3104-3112.

[3] M. Loung, H. Pham, C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," Proc. of EMNL2015, pp.1412-1421.

[4] D.Bahdanau, K.Cho, Y.Bengio, "Neural Mechine Translation by Jointly Learing to Align and Translate," ICLR, 2015.

[5] P. Koehn, "Statistical Machine Translation", Cambrige University Pross, 2009.

C. D. Manning, H. Schutze, "Foundations of Statistical Natural Language Processing," The MIT Press, 1999.

[6] A.Kannan, K,Kurach, S.Ravi, T.Kaufmann, A,Tomkine, B.Miklos,G,Corrado, L. Lukács, M.Ganea, P,Young, V.Ramavajjala, "Smart Reply: Automated Response Suggestion for Email",KDD,2016,pp955-964.

[7] V. Kolesnyk, T. Rocktaschel, S. Riedel, "Generating Natural Language Inference Chains", Arxiv.1606.01404, 2016.

[8] M. Guo, Y. Zhang, D. Zhao, T. Liu, "Generating Textual Entailment Using Residual LSTMs",CCL,2017,pp.263-272.

[9] R. Pasunuru, M. Bansal, "Multi-Task Video Captioning with Video and Entailment Generation",ACL,2017.