Two-way Natural Language Encryption Model(TNLE)

COEN-296: Natural Language Processing

Instructor: Ming-Hwa Wang

Department of Computer Science and Engineering

School of Engineering

Tianxin Zhou

Isaac Wu

Yucheng Chang

Fall Quarter 2020

Abstract

The purpose of this paper is to explore the idea of utilizing a Natural Language Processing (NLP) method to solve a cryptography problem. In the field of cryptography, as the fast development of digit transmission, the encryption and decryption are focusing on mathematically complicating the numbers, so that the security of data transmission can be maximized. However, as new technology is developing, people start considering different ways of encrypting data because decrypting complicated digits becomes much less challenging. In this paper, we will focus on only the text encryption because we consider the process of language translation is very similar to the encryption and decryption of natural language. If we consider our goal is to "translate" the natural language to an encrypted language, some NLP methods can be very well applied to accomplish this goal. One of the most advanced machine translation techniques can be the attention mechanism. We will build a natural language translation model based on the research from Vaswani et al. (2017) and explore the possibility of implementing other NLP techniques including POS tagging or relation extraction.

Table of Contents

List of Figures

1. Introduction

1.1 Objective

This project is about a machine-learning algorithm for encryption and decryption based on the ideas of translating between plaintext and ciphertext. The machine learning algorithm will be based on the improved version of Bert named Albert. The algorithm will take in any passage or article as the “key” to train the model, then it will be able to translate between plaintext to and from the intermediate language or the encrypted form. Without the key or the model, the third party will not be able to retrieve the information that is encrypted.

1.2 Idea originality

There are plenty of methodologies for encryption, and all of them follow the idea of transforming text to another string and then transform it back. The idea intuitively feels like a translation process that translates to and back from some intermediate language that no one could read without a translator. The idea is similar to one who visits a country which uses the language he/she doesn’t understand, so that every word to

him/her seems to be encrypted; Having a translator or a way to look the words up would help the person to decrypt the words and bring him/her a full understanding of them.

A very naive way of encryption was to pick a book and then specify the specific page number, row number, and the kth word of that row. By using this method, students could send a message to another on a paper note to deliver their words. Even if the note is being captured, the teacher will not know what those words' meanings are, and even if they do, they may not know which book to be referred back to. If our algorithm could take any book as a reference, it is possible to add a layer of security into the system.

Combining those two ideas, we want to explore whether it is possible to use a translator that translates the target sentences into an intermediate "language" to do the encryption and decryption of data with reference to a specific article or book. Whether this is a feasible direction, and how well this will perform?

1.3 Relationship with the class

Although this project focuses on machine learning and cryptography at the first glance, the core of it is to translate the natural language to and from an encrypted form. What we are exploring is in fact a translation from one language to "another language" which is the encrypted language. The model we are going to use is the BERT model and its variations, and it is the-state-of-art algorithm in the field of natural language processing. Also, the translation is one of the practical applications of the model. On top of that, pre-processing and feeding an article or book into the model to train it will also require knowledge of NLP.

1.4 Previous works

So far as we researched, there is not much work done in combining the idea of natural language processing with encryption. However, in the translation field, the attention mechanism has been an emerging star with BERT being one significant part of that. However, BERT still has a large number of parameters and may take some time to train and run. A method with a smaller model size would be preferred.

1.5 Our thoughts

With the use of the idea of contents in the book, we could encrypt one word into different forms based on the context of the word, but the translator will know how to translate the word back. With a key from a book, the model would be uniquely trained to translate to and from a unique intermediate language. Also, we will be using the Albert which stands for “a lite BERT” which will try to reduce the number of parameters by sharing parameters between layers and breaking large word embedding matrix. This model will help us to achieve similar results while keeping the model way smaller.

1.6 Area or scope of investigation

We are digging into the area of cryptography and try to find a way to encrypt and decrypt a natural language string. The way we are going to solve this in a natural language processing way. Specifically, we are going to use a transformer based translation model to “translate” between natural language and encrypted Language.

2. Theoretical Background

2.1 A problem of cryptography, NLP, and ML

The problem we are facing is to encrypt the natural language to digits or other form, or decrypt digits to natural language. It is a cryptographic system on natural language level versus the lower level cryptography like encrypting digits during network packet transmissions. Our intention is to use the most advanced machine translation algorithms to “translate” natural language to encrypted messages, and “translate” encrypted messages to natural language. The mechanism needs to go both ways which means we do the “translation” for both encryption and decryption. To be more specific, if we “translate” natural language to encrypted messages with the encoding model, we could then send the encrypted message to the receiver. The receiver could use the encrypted message to feed into the decoding model to get the decrypted messages. In another word, we establish a symmetric encryption method that both parties of a communication will hold a pair of keys. One is the encoder model to encrypt messages. Another is the decoder model to decrypt the message.

Although we’ve tried our best to acquire knowledge about the research regarding the application of the technology in cryptography, NLP, and ML. It appears there is no research that has been done across all three categories. From the cryptography and ML aspect, there is a lot of research focusing on utilizing neural network mechanism to encrypt data. However, because the data nowadays has many forms, including images, text, audios, and so on. It is very difficult for a ML model to learn all different types of data. For cryptography and NLP aspect, there is one research we found that utilizes a rule-based approach based on the semantic and the syntactic meaning of the sentence with the tree structure technique (Jing et al., 2012). For the NLP and ML aspect, we

have been focusing on the statistical approach that utilizes attention mechanism to learn the relationship between words, especially for language translation.

2.2 Attention Mechanism for Machine Translation

In the process of translation, there are two types of layers, one of them is the encoder and another is the decoder. Encoding would include a self-attention, add and normalization, a feed-forward followed by another add and normalization layer. decoding would be a self-attention, an add and normalization followed by a structure that is similar to the encoder layer with slightly different attention patterns. By having those attention mechanisms, the encoder could learn the words' relationship and the context a certain word is in, hence the translator could better understand and turn word from one language to another. In order to add the BERT model into it, we could add another attention that is parallel to the self-attention in the encoder layer, and with the pre-trained BERT model added in, the model could better understand how to encode the word given; in the same way, a BERT attention could also be added to the decoder layer to improve understanding. Both the encoder and decoder are being stacked and each of them takes the output of the previous layer.

2.3 Rule-based Text Encryption

Although encrypting messages on language level seems unnecessary because we could encrypt the data during network transmission, there is research focusing on the language level encryption. The method developed by Jing et al. (2012) is building a dependency tree of the sentence with each word as the leaf. Then, a syntactic or

semantic transformation is applied to the sentence based on the dependency tree. Thus, on the data level, the order of the representation of the words is changed so the sentence will be encrypted.

This approach does apply the fundamental knowledge of NLP. It conceals the original text from the potential attackers. However, more of the application of the method needs to be established so that we could have a better idea of how good the method could be. Instead, if the text to be encrypted gets larger, it may take a very long time to encrypt the whole text. Besides, even if the syntactic and semantic transformation concealed the original text pretty well, it is based on very limited predefined rules which means it is still possible to decipher by observing patterns in a lot of samples.

2.4 Neural Cryptography

One of the fundamental research of applying Neural Network technology with cryptography can be the research by W. Kinzel and I. Kanter in 2002. They developed a synchronizing system which is used to generate keys simultaneously by mutual learning between the parties in a communication. It is an application of ML on data transmission level instead of natural language level. This method should be impossible to be cracked by brutal force with current computer power. Although the technology is not fully developed yet, and we don’t know if the quantum computer will have the power to beat this method yet, it is a potential solution for cryptography in a longer future.

2.5 Natural Language Encryption with Attention

Our research in this paper is to explore the possibility of combining the technology in cryptography, DL, and NLP. As we mentioned in section 2, the text encryption process can be very similar to language translation. Since machine translation is one of the biggest topics in NLP, many advanced DL techniques are developed to solve this problem. Among all newly developed techniques, attention mechanism is proven to be the most effective one.

In addition to machine translation, we could use the traditional book cipher technique to generate the target data which can be one of the oldest cryptographic techniques. The “book”, as the key, can be the combination of different arbitrary or random choices of books or articles. After the input data and target data are generated, the “book”, as the key to decipher, can be discarded or destroyed so the original key cannot be found to decipher the messages. The only possible ways to decrypt will be attaining both the weights and neural network architecture, or brutal force which can be really difficult.

Compared to Neural Cryptography which requires synchronizing mutual learning, our technique will only need to be trained once so the communication cost could be less. Compared to rule-based NLP encryption, our technique is much more difficult to decrypt and easier to be established because neural networks do not require the knowledge of all possible “rules”.

Moreover, one of the biggest challenges for NN machine translation could be learning the semantic meaning for N-gram. In our scenario, we do not need to be concerned about N-gram presentation too much because we do not need the encrypted message to be meaningful, rather than the matching prediction between each “word” in the

original message and the encrypted message. In another word, we only need to consider unigram which could be good enough for our model.

3. Hypothesis (or goals)

In our project(we are proposing), our goal is to build a framework which uses bi-directional encryption model based on a fine-attention model that trains a book into a key book(code) and then uses the trained model as the final keys. In this way, we hypothesize that this could be a breakthrough in data encryption. Since the development of Quantum Computing is going to be the future trend, we consider asymmetric keys based on extremely large prime numbers will no longer be safe.

We assume that in the way after we trained the book and then discard or wiped the book out, the only way to decrypt the ciphertext is to attain our model and the weight we have. Even if the hacker attains the original book, it is still impossible to train out the exact same model as we have. Therefore, we hypothesize that this could be a better approach to solve the conventional encryption problems in the near future.

4. Two-way Natural Language Encryption Model (TNLE)

4.1 Datasets

We will use the Book Cipher method to generate the data to train our model. To be more specific, firstly, we will collect two groups of articles. One group is used as the key, for which we call the key-group. The other group is used to be encrypted by the key, and we call it corpus-group. Secondly, for the key-group, we index every one of the articles or books. For each article or book, we, then, index each sentence. For each word in every sentence, we index them again. Thus, for every word in the key-group,

there is a distinct combination of article-index, sentence-index, and word-index. Thirdly, because the same word may repeat a lot of times in key-group, we collect all distinct words in the key-group and all the combinations for each word. We keep all combinations for each word so we have multiple choices of combinations. In this way, it will be hard to find the patterns in the encrypted messages so the decryption from an attacker may become more difficult. Fourthly, for each sentence in the corpus-group, we collect the combinations for each word in that sentence by randomly choosing a combination for that word in the key-group. In the end, we have the corpus-group as our input for the model, and the collection of the combinations as the target. The whole process is demonstrated in figure 1.

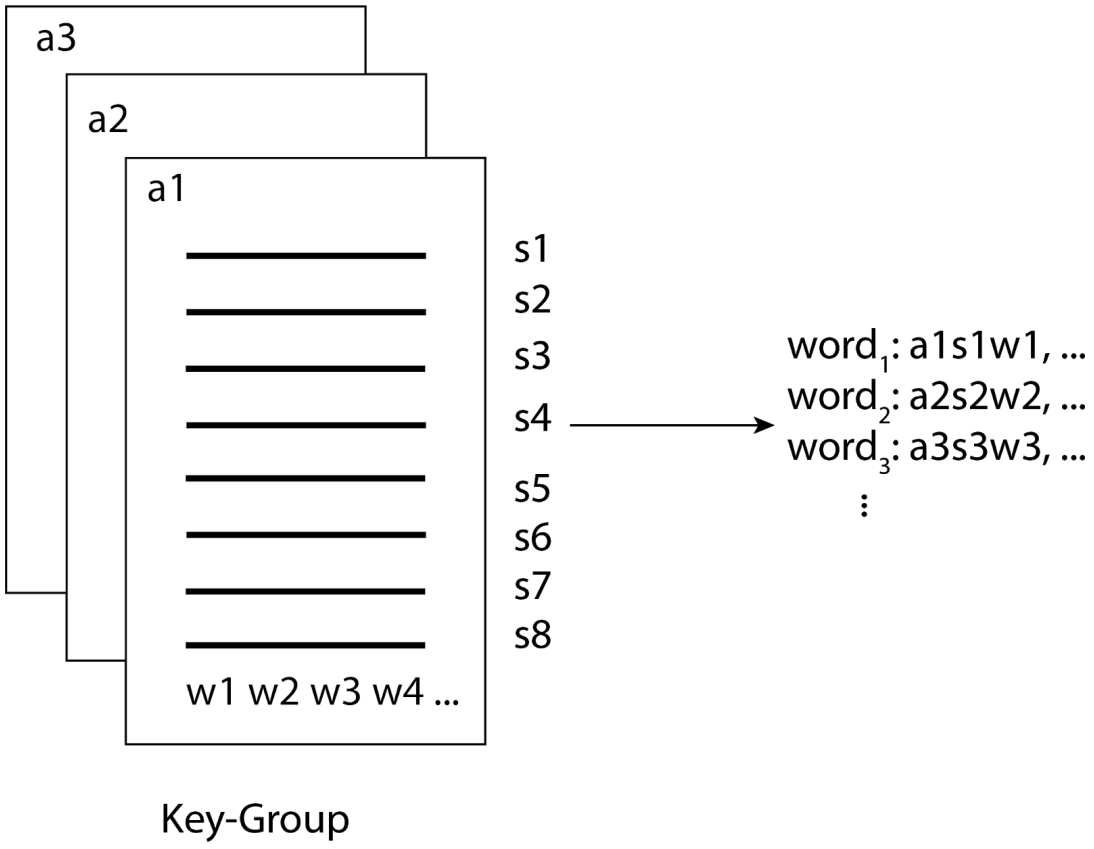

Figure 1. Dataset Generation

4.2 High Level Architecture

In our TNLE model, we will use supervised learning based on attention mechanisms to build a pair of two neural networks. One we call encrypter which is trained and used for encrypting natural language the encrypted messages. The other we call decrypter which is trained and used for decrypting encrypted messages to natural language. Both encrypter and decrypter can have the same neural network structures (could be different). We will use data in natural language as input and paired with the encrypted messages as target to train the encrypter. Reversely, the decrypter will be trained with encrypted messages as input and paired data in natural language as target (figure 2).

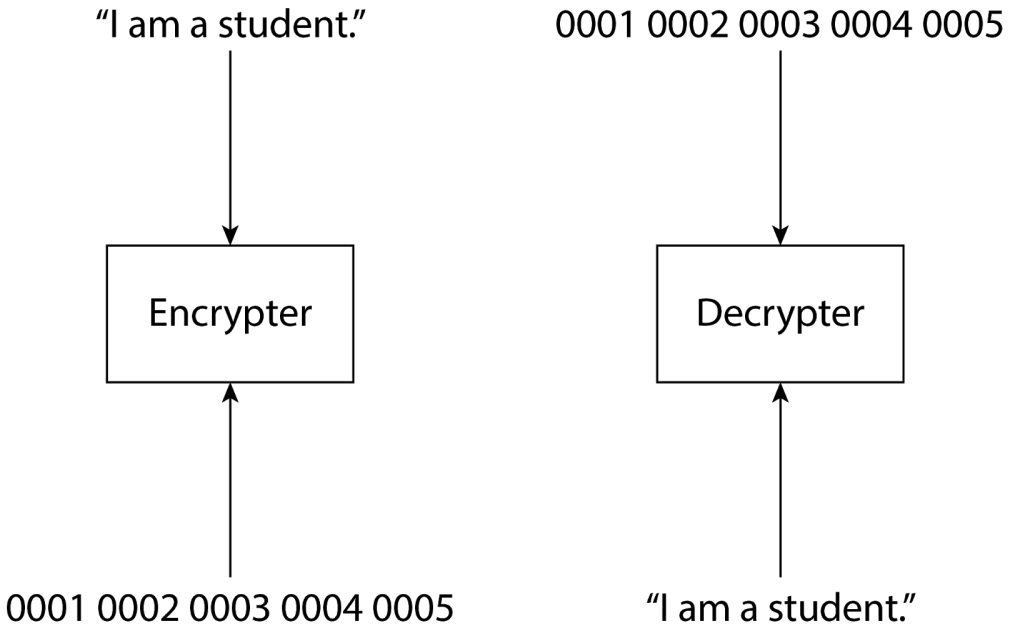

Figure 2. High Level Architecture

4.3 Encrypter and Decrypter - Mid Level Architecture

Both encrypter and decrypter have the same mid-level architecture which is illustrated in Figure 3. The mid-level architecture is a encoder-decoder architecture.

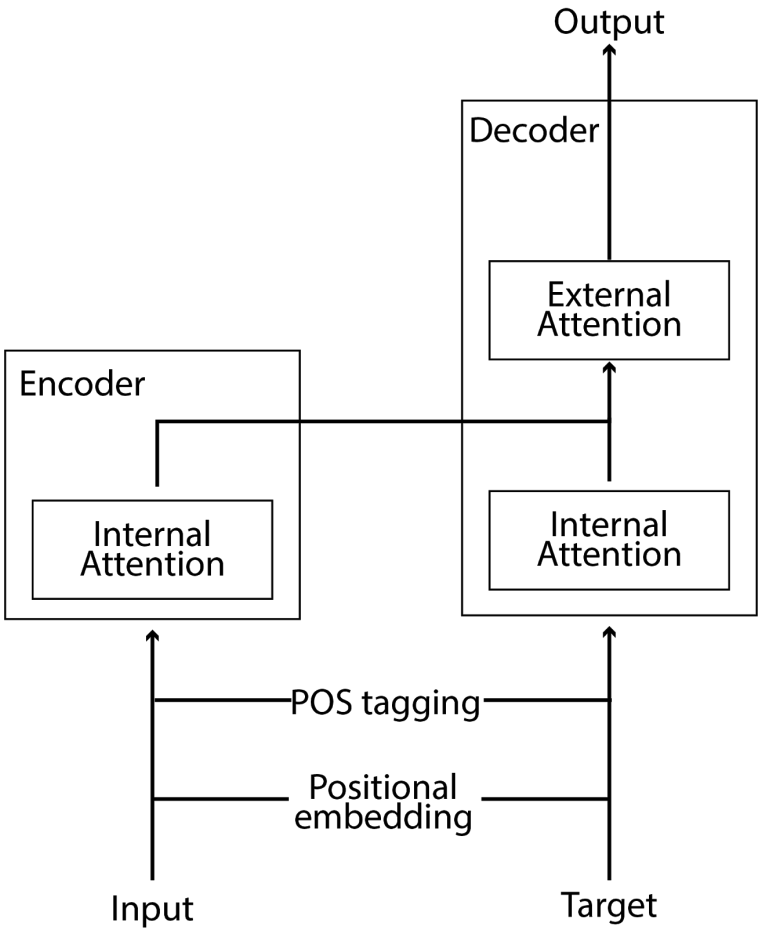

Figure 3 Mid-Level Architecture

4.4 Attention - Low Level Architecture

The attention function (eq.1) is the same as the function from the research of Vaswani et al. By taking the softmax of the dot product of the matrix of query and key, the attention weights can be multiplied by value to get the attention between the elements.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \qquad \text{eq.1}$$

4.4.1 Internal Attention

In the paper from Vaswani et al., the attention between words in the same sentence is called self-attention. We change it to internal attention to better express the difference between the attention of the same sentence and the attention of the different sentence. The internal attention is using same sentence as query, key, and the value, and apply to the equation 1, as figure 4 illustrated.

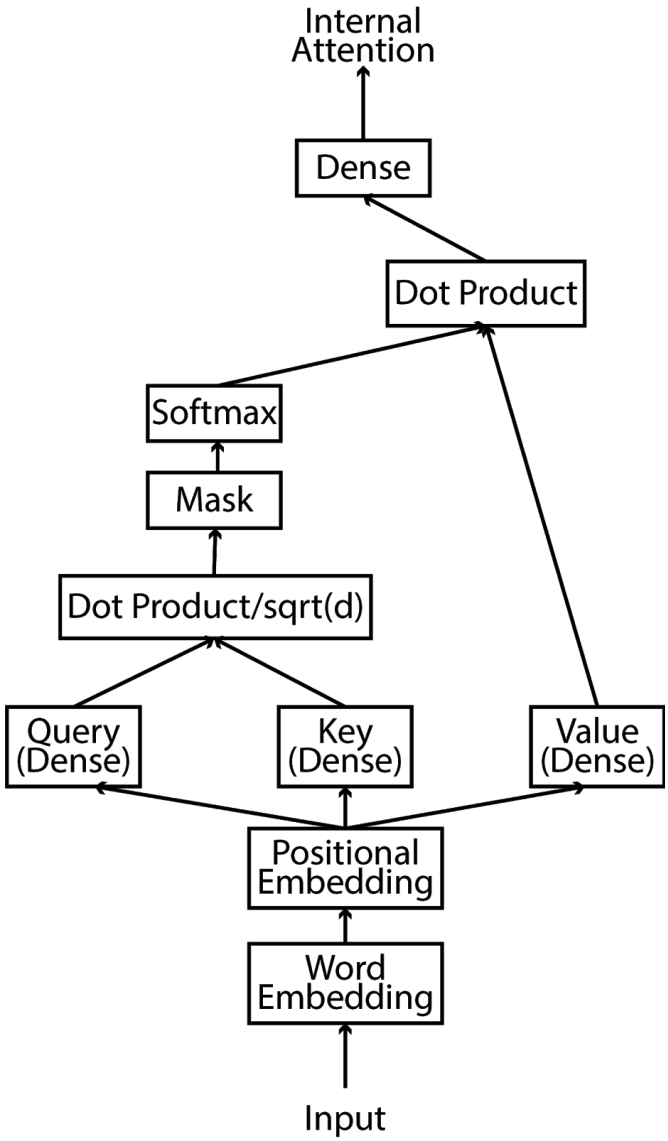

Figure 4 Internal Attention

4.4.2 External Attention

The difference between internal and external attention is the external attention takes the internal attention from the encoder as both key and value, and the internal attention from the decoder for the query. Everything else is the same as internal attention (figure 5).

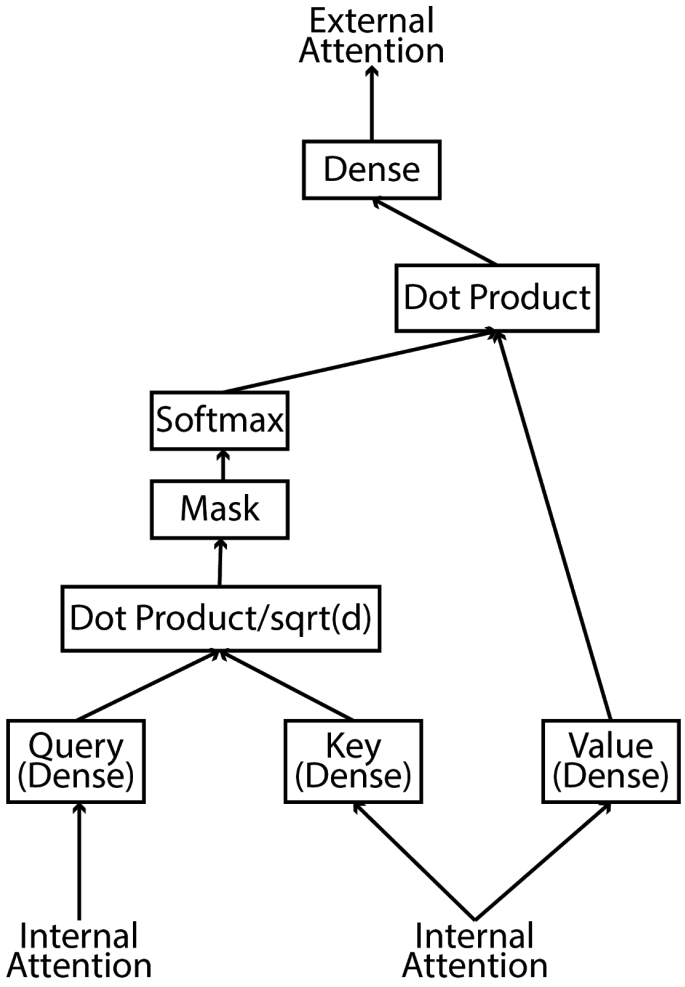

Figure 5 External Attention

The output of external attention is the representation of the attention between each word in different sentences. In machine translation, the difference sentences usually are

the sentences in different languages but have the same meaning. In our project, it represents the attention between words in the input sentence and the combinations in the target encrypted sequence.

4.5 Programing Language and Tool

We are going to use python to build our program with, majorly, the tensorflow library. Some neural network layers, like the dense layer, will be from keras library. However, keras is based on tensorflow, so it is essentially only tensorflow as the backbone.

The tool we use is the Google Colab. Not only does it provide 12-hour GPU usage, but also it is a cloud platform so all of us can access and edit our code pretty conveniently.

4.6 Encryption Output

After the model is trained, first for the encrypter, we can take any sentence in natural language as input for the encoder and an empty sequence as the target to feed into the decoder. Then, the model should predict the first combination and we add it to the empty sequence as the new target to generate the next combination until the end-of-sequence or the max-length is reached.

Then, we take the output of the encrypter which is a sequence of combinations as the input of the encoder for the decrypter. An empty sequence is fed to the decoder of the decrypter. We can have the first predicted word and add it to the empty sequence again as the target for the decoder to predict the next word.

In the end, we will have the sequence of combinations from the encrypter and the sequence of words from the decrypter. The expectation is the output of the decrypter to be the same as the original input sentence.

4.7 Evaluation

After we generate the dataset as mentioned in 4.1, we will divide the corpus-group into train set and test set. We will only use the train set for the training. To evaluate the performance of the model, we will use the test set. For each sentence from the test set, we compare it with the output of the decrypter and calculate the mean accuracy as one of the assessments of the model's performance.

The other assessment could be visualizing the attention from the low level architecture. For the encrypter, we expect to see the attention between input words and the combinations. If we plot the attention between combinations and output words from the decrypter, if we compare the two, we expect to see similar attention between the same pair of combination and word.

5. Implementation

Bibliography

Jing, X., Hao, Y., Fei, H., & Li, Z. (2012). Text Encryption Algorithm Based on Natural Language Processing. *2012 Fourth International Conference on Multimedia Information Networking and Security*, 670-672. 10.1109/MINES.2012.216.

Kinzel, W., & Kanter, I. (2002). Neural Cryptography. *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02*, *3*, 1351-1354. 10.1109/ICONIP.2002.1202841

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All You Need. *CoRR*, *abs/1706.03762*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

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

Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., ... & Liu, T. Y. (2020). Incorporating bert into neural machine translation. arXiv preprint arXiv:2002.06823.